

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Diseño de un sistema para el manejo a distancia
de aeronaves construidas con arduino

Autor: GenLei Cui Ji

Director: Víctor Nieto Lluís

MADRID, JUNIO DE 2014

“Educad a los niños y no será necesario castigar a los hombres”

Pitágoras de Samos

“El sabio puede cambiar de opinión. El necio, nunca”

Immanuel Kant

AGRADECIMIENTOS

En primer lugar, quisiera agradecer de corazón toda la ayuda, guía y consejo que me ha proporcionado mi tutor, Víctor, sin los cuales este proyecto nunca habría tomado forma. Los momentos más críticos fueron al comienzo de este proyecto, eché a volar mi imaginación, pero Víctor, me puso los pies en el suelo y me orientó hacia el rumbo adecuado. En definitiva, me ayudó a enfocar el proyecto hacia un objetivo concreto y realizable. Mis más sinceros, cordiales y respetuosos agradecimientos, Víctor.

En segundo lugar, me gustaría poder ofrecerles algo tan valioso o más como la oportunidad que me han brindado mis padres para educarme y prepararme para el futuro profesional de la mejor forma que han sabido, podido y luchado, que hoy en día muy pocos lo valoran como es debido. Sé que ellos lucharon toda su vida por ofrecerme esta oportunidad y espero que como recompensa puedan sentirse orgullosos de mí algún día. Gracias Mamá y Papá por todo.

En tercer lugar, No quisiera olvidarme de mis compañeros y amigos que han estado a lo largo de todo el semestre interesándose por mi proyecto, haciéndome la pregunta de “*pero, ¿Ya vuela?*”. Me gustaría agradecerles por el apoyo y en muchas ocasiones también la ayuda recibida de ellos. Muchísimas gracias Javi, Iñigo, Caba, Adrián, Ángel, Jorge, Jelle, Raúl, Manu, Victoria, Luis, José y muchos otros que nuestros caminos se cruzaron por la universidad y por la vida. Gracias a todos por haber compartido todos estos años de alegría y sufrimiento.

Por último, tampoco quisiera olvidarme de todo el profesorado que he ido teniendo a lo largo de todos estos años por la Facultad de Informática, les agradezco por todas las ayudas y conocimientos que me han aportado y en especial me gustaría agradecer de corazón a Ángel Lucas González, Loïc Martínez, Ricardo Imbert, Francisco Rosales, Antonio Pérez, Fernando Pérez, Pedro de Miguel y María Luisa Córdoba.

ABSTRACT

The aim of this project is to attempt to reduce the costs of some surveillance services offered by security agencies or institutions.

The project consists in designing and developing a computer system to remotely control a drone or quad-copter through a computer, manipulating the drone through the keyboard and watching the images captured from the camera module. Each drone is built with one or more modules, and each module has its own functionality. Both new drones and new users can be added to the computer system through the drone management system and the user management system, respectively. Both of management systems are going to be developed. The project also includes the making of a quad-copter prototype and a controller unit implementation.

RESUMEN

El principal objetivo del proyecto es intentar reducir los costes de algunas de las operaciones de vigilancia llevadas a cabo por agencias o instituciones de seguridad.

El proyecto consiste en diseñar y desarrollar un sistema informático que permita el manejo a distancia de un cuadricóptero a través de un ordenador, utilizando el teclado y visualizando las imágenes recibidas del módulo de la videocámara. Cada cuadricóptero estará compuesto de diferentes módulos y cada módulo tiene una funcionalidad característica. Se desarrollará un sistema de gestión de aeronaves para poder añadir nuevas unidades de cuadricópteros, así como un sistema de gestión de usuarios para administrar los usuarios en el sistema. Adicionalmente, se construirá un prototipo de cuadricóptero y se implementará su unidad controladora para poder realizar las pruebas del sistema desarrollado con ello.

Índice

Introducción.....	vii
Estado de arte	vii
Capítulo 1: Memoria técnica	1
Capítulo 2: Requisitos del sistema	6
Capítulo 3: Diseño de alto nivel	7
Capítulo 4: Diseño de bajo nivel	8
Capítulo 5: Implementación	9
Capítulo 6: Pruebas.....	11
Capítulo 7: Normativas.....	14
capítulo 8: Licencias	15
Capítulo 9: Conclusiones.....	16
Capítulo 10: Líneas futuras	17
Bibliografía.....	18
Anexo A: Memoria Técnica	20
Anexo B: Especificación de Requisitos del Sistema.....	21
Anexo C: Diseño de Alto Nivel.....	22
Anexo D: Diseño de Bajo Nivel.....	23
Anexo E: Manual de instrucciones	24

INTRODUCCIÓN

El presente proyecto desarrollará un sistema de comunicación, que permitirá a los usuarios manejar una aeronave a distancia, consistente en una interfaz web, una base de datos, un esquema XML, un servidor de comunicaciones y un prototipo de cuadricóptero con su unidad controladora.

En el ámbito educativo, este proyecto tiene como objetivo el estudiar y desarrollar las tecnologías que hacen posibles el sistema anterior. En el ámbito de la aplicación práctica, se pretende utilizar el sistema para la reducción de costes de algunas operaciones de vigilancia, llevadas a cabo por agencias o instituciones de seguridad.

ESTADO DE ARTE

Al iniciar el presente proyecto, se ha tomado como referencia dos trabajos desarrollados también en el entorno educativo y que guardan mucha similitud con las tareas a realizar en ésta.

El primero es el proyecto fin de carrera realizado por Carlos Ortega Solé [1]. Éste consiste en el desarrollo de un sistema virtual que permite la visualización de un túnel durante el vuelo, que representa la trayectoria de la misión planificada para una aeronave no tripulada, en adelante UAV (en inglés, *Unmanned Aerial Vehicle*).

El proyecto de Carlos consiste en utilizar señales de radiofrecuencia 2,4GHz para las comunicaciones entre la emisora y la UAV, además de utilizar otra señal de radiofrecuencia de 5,8GHz para un sistema de visionado en primera persona, en adelante FPV (en inglés, *First-Person View*). Además, utiliza otra señal de radiofrecuencia adicional a las anteriores para la transmisión de telemetría, que por cuestiones legales se ha utilizado la frecuencia de 433MHz.

En cuestiones de software, ha decidido utilizar el software programador de misiones *Mission Planer*, que además le proporciona firmwares ya desarrollados para el cuadricóptero construido en su proyecto. En cuanto a la hora de realizar la virtualización de los túneles ha preferido utilizar la herramienta *Unity3D*, que le permite recrear la parte gráfica necesaria de virtualización.

Por último, en cuestiones de comunicación ha utilizado el protocolo MAVLink, que está basado en el estándar CAN bus y SAE AS-4 sobre los paquetes de comunicación.

El segundo proyecto encontrado es el de Xabier Legasa Martín-Gil [2], que opta por diseñar un sistema de manejo remoto basado en *Android* y controla la unidad de cuadricóptero a través de un módulo *Bluetooth*.

El proyecto utiliza también la arquitectura de plataforma hardware libre Arduino. El entorno de desarrollo escogido es el de Android para la interfaz de usuario, en cambio, para la unidad controladora utiliza el entorno de desarrollo Arduino.

Como se ha podido observar en el proyecto [2], la utilización de varias señales en diferentes espectros de radiofrecuencia ha obligado a tomar muy en cuenta las regulaciones legales que cada país tiene en esa materia, por tal motivo en este proyecto se optó por utilizar la tecnología WiFi, que utiliza un espectro de radiofrecuencia internacionalmente válida [21]. Tanto esta tecnología como la de Bluetooth presentan una seria limitación en cuanto a la distancia de cobertura. No obstante, la tecnología WiFi presenta la ventaja de poder extender el radio de cobertura de la señal por medio de estaciones repetidoras.

Ambos proyectos han escogido utilizar la plataforma de hardware libre Arduino, lo cual ha motivado el que haya sido seleccionado también para el desarrollo del presente proyecto, dado que los componentes oficiales de Arduino son de un tamaño considerable para algunos diseños de cuadricóptero, como se puede apreciar en la [Figura 29](#) en el [Anexo A: Memoria Técnica](#), se ha preferido hacer uso de un nuevo proyecto basado en Arduino, pero con dimensiones físicas del hardware mucho más reducidas, denominado Microduino [3]. Las placas de Microduino son mucho más pequeñas y modulares permitiendo así conectarse entre diferentes módulos sin necesidad de soldaduras, *Shields* o cables.

La interfaz de usuario en los dos proyectos tomados como referencia resulta ser bastante limitada en cuanto a su portabilidad, por lo que en este proyecto se desarrollará una interfaz web, que tiene soporte en todos los sistemas operativos y dispositivos que contengan un navegador de páginas web.

En cuanto al ámbito comercial existen modelos de cuadricópteros que ya están disponibles en el mercado, tales como el *AR.Drone* de [6], se puede apreciar el producto en la [Figura 1](#), que tienen unas especificaciones muy similares al sistema a desarrollar en el presente proyecto. *AR.Drone* utiliza las señales WiFi para establecer las comunicaciones con el cliente, pero el cliente sólo está disponible para plataformas móviles (Android Os y iOS). Adicionalmente, el aparato actúa de servidor de la señal WiFi, por lo que tiene cierta limitación o restricción de uso delimitado por la cobertura de la señal WiFi del aparato.



Figura 1: AR.Drone

En cambio, el aparato puede retransmitir unas imágenes de alta calidad en tiempo real, además de ofrecer informaciones de vuelo, también en tiempo real. El modelo ofrece además, algunas configuraciones o personalizaciones posibles, tales como intercambiar el tipo de batería por alguno de mayor capacidad, personalizar la estructura del chasis y diferentes funcionalidades digitales como programación del plan de vuelo o compartir información entre usuarios, entre otras existentes.

Existe otro modelo en el mercado, que es el *Phantom* de [4], como se puede apreciar en la [Figura 2](#). Éste ofrece una comunicación por radiofrecuencia (en adelante RF) controlada por un mando de control, lo cual obliga al usuario final estar atado en una zona física dada, para poder emplear la aeronave en cuestión. Pero este producto ofrece funcionalidades de gran interés como las que tiene el AR.Drone, programar el plan de vuelo, dado por la tecnología de RF, se puede programar con una distancia mayor que la de AR.Drone. Tiene una funcionalidad que es auto detectar si se ha salido del perímetro de cobertura máxima y si es así, el cuadricóptero regresa automáticamente hasta el punto de inicio (llamado, en inglés, *Return To Home*).



Figura 2: DJI Phantom

Los modelos comerciales encontrados tienen unos costes iniciales a partir de los 249,99\$ en el caso de AR.Drone y a partir de los 589€ en el caso de Phantom.

CAPÍTULO 1: MEMORIA TÉCNICA

En la construcción del prototipo de aeronave (mostrado en la [Figura 3](#)), se ha escogido un diseño específico, correspondiendo a un cuadricóptero. Todo lo relativo a los materiales y componentes empleados, así como los pasos seguidos en su construcción, se explican detalladamente en el [Anexo A: Memoria Técnica](#).



Figura 3: Prototipo cuadricóptero construido

Sin duda, los componentes más significativos en el diseño del prototipo lo constituyen los módulos Microduino (mostrado en la [Figura 4](#)) y la videocámara inalámbrica Ai-Ball (mostrado en la [Figura 5](#)). Ambos componentes son controlables con la tecnología WiFi, lo cual permitirá unificar las comunicaciones por medio de un mismo router y pudiendo ampliar su radio de cobertura utilizando estaciones repetidoras.

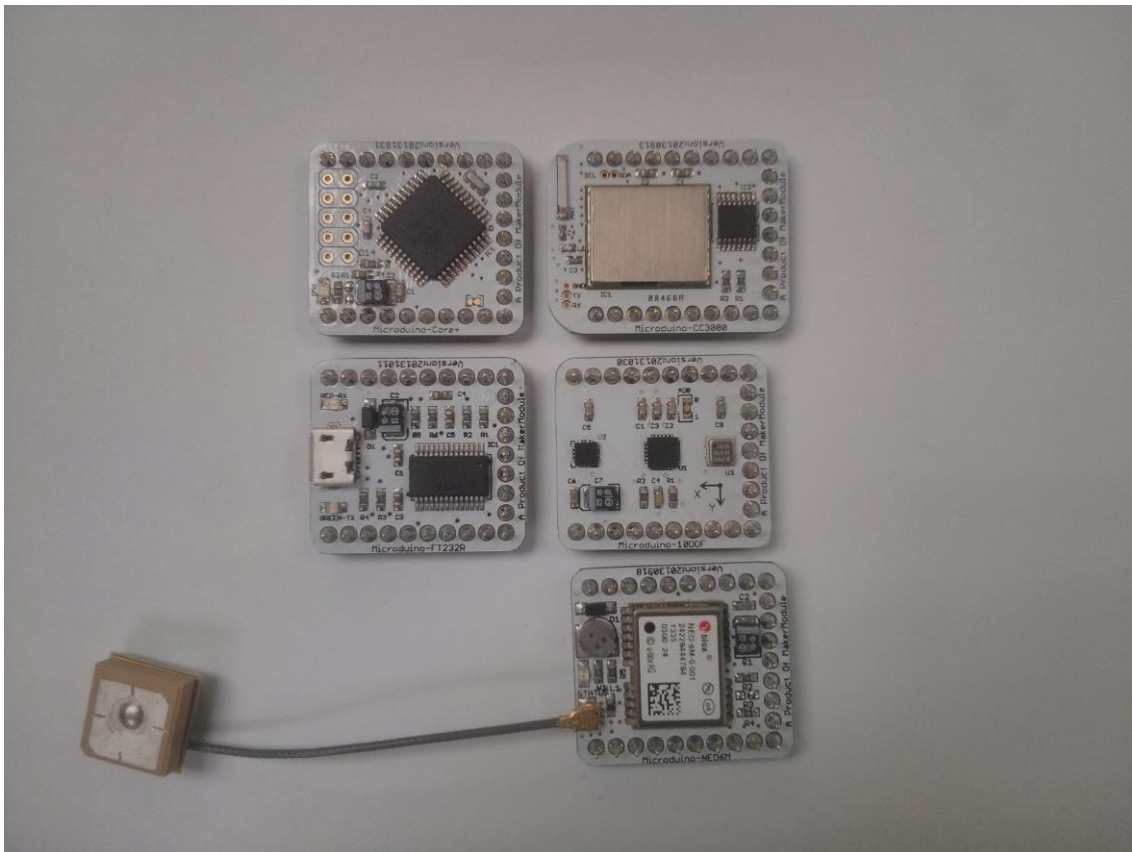


Figura 4: Módulos Microduino



Figura 5: Videocámara inalámbrica Ai-Ball

El coste de las adquisiciones de todos los componentes ha sido aproximadamente de 354,70€ y se repartieron entre diferentes proveedores buscando siempre a los que ofrecían el precio más competitivo. Se puede consultar detalladamente el listado completo de todos los elementos adquiridos el documento [Anexo A: Memoria Técnica](#).

Durante la construcción del cuadricóptero surgieron varios contratiempos imprevistos, algunos de los más relevantes se comentan a continuación:

1. Hubo retrasos en las entregas de algunos componentes esenciales para poder finalizar la construcción del cuadricóptero. Tras varios intentos de comunicación con el proveedor, no hubo otra opción que la de esperar hasta recibirlos. Esto conllevó cierto retraso en la finalización de construcción del prototipo.
2. Se encontró con que no existía un soporte donde emplazar los módulos Microduino adecuadamente al chasis del prototipo. Se soluciona gracias a que con las adquisiciones venía un módulo Microduino-Vduo, que resuelve perfectamente este problema, ayudando además reducir la altura de la instalación de los módulos Microduino necesarios en el prototipo, como se puede observar en la [Figura 6](#).
3. El prototipo necesitaba aprovechar algunos puertos extras que no venían preparados por defecto en el Microduino-Core+, por lo que fue necesario realizar algunas soldaduras adicionales entre el módulo Microduino-Core+ y Microduino-Vduo, como lo mostrado en la [Figura 7](#).

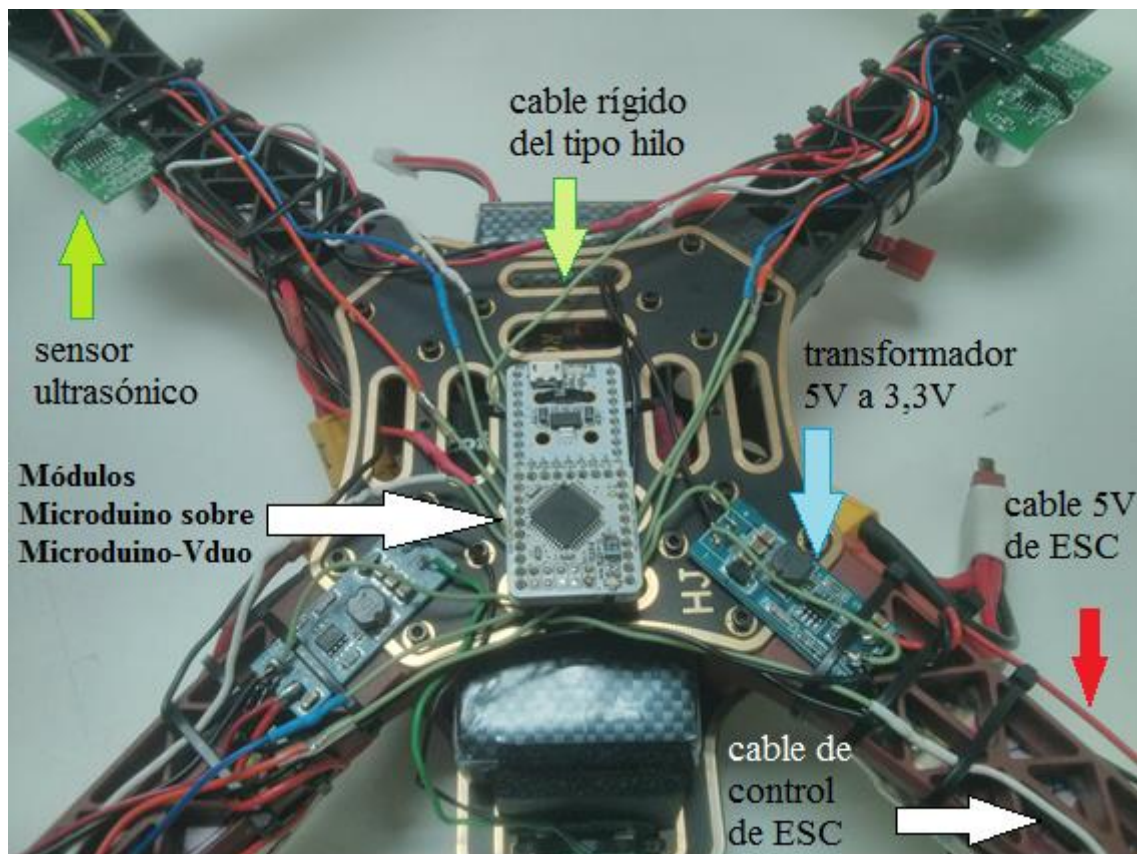


Figura 6: Componentes instalados en el prototipo



Figura 7: Unión de puertos entre Microduino Core+ (arriba) y placa V-duo (abajo)

Los problemas anteriormente mencionados son relativos a la construcción de la aeronave, pero existen otras cuestiones que son de vital importancia y que tienen que ver con los factores de riesgo que la propia aeronave pueda generar.

La principal causa de la peligrosidad del prototipo de cuadricóptero son las hélices de cada motor, que están al aire libre sin ningún tipo de protección. Los motores utilizados en la construcción del prototipo son del tipo *brushless*, que por definición son motores sin escobillas, lo cual les permiten girar con bastante más velocidad, dado que existe menor rozamiento y por ende mayor rendimiento.

Una posible solución a este problema es utilizar protectores complementarios que rodeen físicamente las hélices, evitando así cualquier contacto físico posible con ellas, en la [Figura 8](#) se puede observar un ejemplar. Al mismo tiempo se pueden complementar con sensores de proximidad y de choque, ayudando a desacelerar o parar los motores en caso de detectar estar cerca o haberse chocado contra algo.

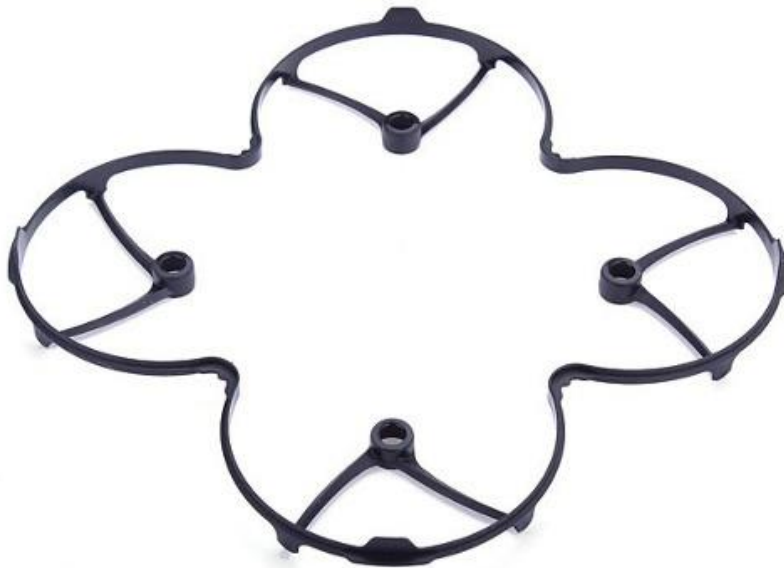


Figura 8: Protección contra las hélices

Sin embargo, por las limitaciones en los presupuesto del proyecto se ha optado por otra solución, que es evitar el uso del prototipo en zonas desprotegidas y concurridas.

Concretando, en la parte técnica que corresponde a la primera fase del proyecto se ha desarrollado con total normalidad. Los problemas que surgieron han sido fácilmente resueltos y se ha dedicado aproximadamente 20 horas en esta primera fase, sin contabilizar la dedicación a las correcciones de la documentación a posteriori.

CAPÍTULO 2: REQUISITOS DEL SISTEMA

Previo al comienzo del desarrollo del proyecto, se ha definido una serie de *features* o funcionalidades que el futuro sistema deberá disponer y cumplir. Para ello se ha redactado el [Anexo B: Especificación de Requisitos del Sistema](#), en el cual se especifica el ámbito y el alcance del sistema a desarrollar, los requisitos iniciales, requisitos no funcionales, 28 requisitos que son agrupados en 4 funcionalidades diferentes y se ha establecido que el nombre del sistema a desarrollar será SIMACA, un acrónimo de “Sistemas de manejo a distancia de aeronaves construidas con arduino”.

El entorno de ejecución del sistema está ambientado en una oficina. Lugar donde están los usuarios finales que podrán utilizar sus equipos informáticos para acceder a un servicio web, que es el presente sistema a desarrollar. Adicionalmente, los mismos usuarios podrán manejar una aeronave registrada en el sistema. Estas aeronaves podrían estar localizadas en otras zonas geográficas. Los usuarios podrán manejar con el teclado de sus equipos las aeronaves, igualmente podrán observar las imágenes capturadas por los módulos de videocámara instaladas en dichas aeronaves en los monitores de los equipos informáticos, para realizar una tarea a distancia con las aeronaves.

Uno de los requisitos iniciales esenciales para el correcto funcionamiento del sistema es la disponibilidad del uso de las tecnologías WiFi 802.11b/g, dado que los módulos operan bajo dicha tecnología para comunicarse con el sistema a través de un router WiFi.

Una de las características principales del sistema es que es primordial el registro de los usuarios, para reconocer si los usuarios finales son pilotos cualificados y aptos para manejar las aeronaves a través del sistema a desarrollar.

Las 4 funcionalidades principales se resumen en que se debe gestionar la parte de acceso, registro y gestión de los usuarios; el sistema debe facilitar comodidades y funcionalidades para las operaciones de manejo de cada aeronave registrada en el sistema; dado que el sistema prevé el manejo de varias aeronaves, es necesario que se contemple un servicio de gestión de dichas aeronaves (altas, modificaciones y bajas) y por último, dado que cada aeronave tiene sus particularidades, el sistema debe facilitar la personalización o configuración personalizada de cada aeronave.

Los puntos mencionados están explicados detalladamente en el documento [Anexo B: Especificación de Requisitos del Sistema](#).

CAPÍTULO 3: DISEÑO DE ALTO NIVEL

En cuanto al diseño de alto nivel para el proyecto, se ha decidido realizarlo mediante el uso acontecimientos y burbujas DFD. Adicionalmente se incluye en este apartado el diseño de la base de datos definido según el uso de los diagramas de entidad-relación y DD. Por último, se define también la unidad controladora del cuadricóptero, así como el protocolo de comunicación entre las aeronaves y el servidor del sistema.

Cabe destacar que el funcionamiento básico de sistema, queda reflejado tal y como se muestra en la [Figura 9](#). Existirá al menos un administrador que podrá supervisar el registro y gestión de los usuarios, mientras que los usuarios dados de alta podrán manejar las aeronaves registradas en el sistema y por último, las aeronaves registradas pueden comunicar con el sistema para aportar informaciones de vuelo.

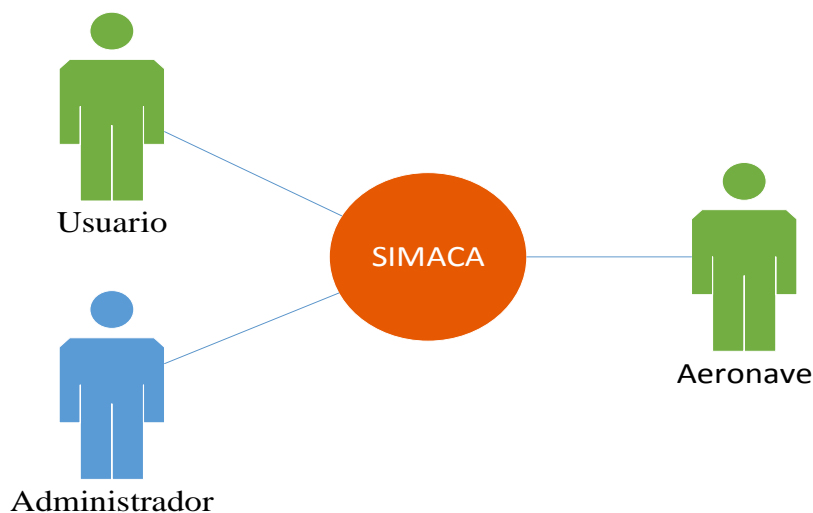


Figura 9: Diagrama de funcionamiento básico del sistema

El diseño de alto nivel ha sido desarrollado en el documento [Anexo C: Diseño de Alto Nivel](#), se debe consultar dicho documento para conocer más detalles sobre las decisiones tomadas.

CAPÍTULO 4: DISEÑO DE BAJO NIVEL

En esta parte del desarrollo se han creado las tablas de la base de datos y se ha definido utilizando algoritmos y pseudocódigo, el desarrollo de cada proceso o acontecimiento y la unidad controladora del cuadricóptero. Todos los detalles relativos al diseño de bajo nivel podrán consultarse en el documento [Anexo D: Diseño de Bajo Nivel](#).

No obstante, tras el desarrollo del primer ciclo, se ha preparado un manual de instrucciones, que explica desde cómo se prepara el entorno de instalación del servidor, la propia instalación del servidor, la configuración inicial del servidor, la instalación del sistema implementado hasta este primer ciclo y el uso del sistema desarrollado. El contenido íntegro del manual se podrá consultar en el documento [Anexo E: Manual de instrucciones](#).

CAPÍTULO 5: IMPLEMENTACIÓN

Durante la implementación del código se han tenido en cuenta 2 partes, la parte de análisis y estudio y la parte de codificación.

En cuanto a la parte de análisis y estudio, es la que más tiempo ha requerido en este primer ciclo del proyecto, dado que se va a intentar establecer una comunicación desde una página web abierta en cualquier navegador y pueda establecer una comunicación directa con un servidor Arduino, localizado en una aeronave.

En un principio, existen limitaciones sobre el tipo de comunicación perseguida por el proyecto, dado que la única forma existente para una comunicación en tiempo real entre un navegador y un servidor es el uso de las APIs ofrecidas por los servicios web *RESTful* y otras similares, pero esta solución resulta ser una carga relativamente pesada de ejecutar para el servidor, puesto que se necesita un entorno con una capacidad de procesamiento y almacenamiento alto, además los envíos de las cabeceras HTTP son bastante extensas, comparado con los entornos de ejecución Arduino utilizado por este proyecto. Por tanto, se ha buscado una alternativa, que es el uso de las *websockets* del borrador de HTML5 llevado a cabo por [5]. Adicionalmente, los servidores socket para Arduino son bastante livianos y se puede comunicar con simples mensajes, esto ayuda a reducir la carga y la capacidad necesaria.

Sin embargo, la definición de *websockets* y *sockets*, no es exactamente igual, porque *websockets* está basado en el protocolo HTTP/1.1 [7], que gracias a su característica de las sesiones, permite reducir la cantidad de cabeceras a enviar en una conexión establecida, lo cual permite enviar el mensaje en una sola línea, para que un servidor que esté escuchando desde el *websocket* definido, interprete dicha línea recibida y pueda realizar una acción determinada. En cambio los *socket* funcionan a otro nivel, que se adecua más al entorno de redes que de protocolos, por lo que no se entienden entre ellos, motivo por el cual se ha estudiado y decidido implementar un servidor intermediario que hará de interfaz de comunicación entre la página web y la aeronave arduino.

En cuanto a la parte de codificación, se han tenido muchos problemas para desarrollar un servidor y cliente basado en *websocket*, debido a que es una tecnología novedosa. No obstante, se han encontrado varios tutoriales [8] [9] [10] [11] y ejemplos desarrollados que resultaron útiles como punto de partida para el desarrollo del servidor y cliente desarrollado para el sistema.

Por otro lado, hubo bastantes complicaciones a la hora de implementar el código de la parte de Arduino, puesto que es una parte que depende mucho de los componentes electrónicos instalados en la aeronave y ello requirió en muchas ocasiones realizar pruebas empíricas para establecer algunos valores propios de cada componente y que éstos deberían de haber sido facilitados por el proveedor. Adicionalmente, el proyecto trabaja con los módulos Microduino, que es un proyecto de reciente creación y todavía no cuentan con un amplio abanico de módulos, por lo que se está desarrollando con cierta limitación en cuanto a componentes electrónicos y es por ello que se limita, a veces, el uso de determinadas librerías disponibles para el desarrollo.

En resumen, la implementación del código no ha resultado ser nada sencilla, se ha requerido invertir mucho más tiempo en el estudio y análisis previos que lo invertido propiamente en la implementación del código.

CAPÍTULO 6: PRUEBAS

La finalización de este primer ciclo de desarrollo del proyecto ha permitido realizar varias pruebas sobre el sistema en conjunto.

Se ha podido realizar una instalación completa de los requisitos básicos para el funcionamiento del sistema desarrollado, es decir, se ha dispuesto de una aeronave, un router WiFi, un servidor intermediario y un equipo con un navegador.

Las pruebas realizadas se reducen a las maniobras de elevar y descender, para comprobar que la aeronave puede despegar correctamente.

Durante las pruebas iniciales, se ha comprobado que la parte del cliente funciona debidamente, pero en ocasiones la comunicación entre el módulo de la cámara y el cliente no se establecía. Esto se debió a que la cámara, mientras funcionan los motores de la aeronave, recibe muchas vibraciones y esto afecta a las soldaduras de los contactos, que con las reiterativas pruebas se agrietaron y originaron falsos contactos.

En cuanto al servidor intermediario, establece correctamente las conexiones *websockets*, pero a veces produce algunos avisos de operación no reconocida, estos avisos no afectan en absoluto el funcionamiento de la aeronave, pero no se ha intentado averiguar el origen de los avisos.

En lo referente al servidor localizado en la aeronave, podemos decir que éste funciona correctamente salvo en algunas ocasiones en las que, durante el arranque de los motores, uno de ellos se queda inmóvil, es decir, no inicia el movimiento giratorio. Este fallo se resuelve reiniciando la aeronave. Se desconoce con exactitud su origen, pero se duda del funcionamiento de los ESC, debido a que el proveedor no ha facilitado una detallada información sobre las características de su funcionamiento.

Por todo ello, a pesar de que todo el sistema implementado parece funcionar adecuadamente, según los resultados de los test a los que se le ha sometido, la aeronave no consigue despegar controladamente. En las pruebas realizadas, la aeronave comienza a girar sobre sí misma hacia un lado u otro, descontroladamente, como se puede apreciar en la [Figura 10](#), [Figura 11](#) y [Figura 12](#). Pero recibe correctamente las órdenes enviadas por el usuario final.



Figura 10: Despegue desincronizado 1



Figura 11: Despegue desincronizado 2



Figura 12: Despegue desincronizado 3

Se ha intentado averiguar la causa del problema e inicialmente se pensó que era el problema de uno de los motores, porque funcionaba a diferente velocidad y, además, transmitiendo una vibración claramente mayor al del resto de los motores. Como solución, se procedió a sustituirlo por otro cuyo comportamiento fuera más parecido al de los otros tres motores. Tras reanudar las pruebas después de la sustitución del motor, se observa una pequeña mejoría, pero la aeronave seguía girando hacia un lado u otro incontroladamente. La causa puede residir en las características de los controladores de velocidad electrónicos (Siglas en inglés, ESC), responsables de asegurar las características dinámicas que debe entregar cada motor en respuesta a una demanda concreta, y la solución en un principio habría que buscarla a nivel de software, ajustando independientemente cada ESC. En las pruebas de vuelo realizadas, se partió de la base de aplicar un rango de frecuencias de funcionamiento común para los 4 ESC (misma marca y modelo), pero se pudo observar claramente la descompensación dinámica existente entre los cuatro motores, lo que provocaba un vuelo totalmente inestable y de difícil control. La dificultad para resolver este problema seguirá residiendo en cómo ajustar y sincronizar las frecuencias para que los motores entreguen la misma potencia en las maniobras de despegue y de vuelo estabilizado.

CAPÍTULO 7: NORMATIVAS

Coincidiendo en el tiempo con el desarrollo de este proyecto, la Agencia Española de Seguridad Aérea (AESA) hizo público un documento [\[12\]](#) en el que se hacían una serie de declaraciones sobre las normas de uso de diversos tipos de aeronaves. En este documento se hace una clara definición de qué aeronaves son consideradas drones y cuáles son consideradas aeronaves.

Según este mismo documento normativo, está terminantemente prohibido hacer volar los drones, categoría a la que pertenecerá el cuadricóptero de este proyecto, dado que está orientado a un uso profesional, en espacios abiertos por falta de una normativa específica que regule la operación con este tipo de aeronaves. Sólo se permite su uso en espacios acotados y sin peligro para las personas.

La reglamentación actualmente vigente reconoce a este tipo de aeronaves como uno más, por lo que se les aplica la misma normativa en cuanto a revisión, registro y certificación para poder obtener una licencia y matrícula de vuelo, lo mismo que se le exige a un avión tripulado.

En estos momentos ya se está trabajando en el desarrollo de una regulación normativa que servirá para clasificar y certificar a este tipo de aeronaves no tripuladas y operadas por control remoto. Algunas de ellas podrían quedar exentas de la necesidad de su matriculación y de certificación de navegabilidad, pero sí quedarían claramente especificados los requisitos mínimos de seguridad que deben cumplir en su fabricación, mantenimiento y operación.

Provisionalmente, hasta que la nueva normativa no quede aprobada y publicada, se podrán autorizar licencias de uso temporal para poder utilizar este tipo de aeronaves no tripuladas con fines demostrativos o de prueba, siempre que hayan sido solicitadas previamente a la autoridad competente para ello, la AESA.

CAPÍTULO 8: LICENCIAS

En la realización de este primer ciclo del proyecto han sido utilizados componentes y servicios de terceros, lo cual nos ha obligado a hacer uso de diferentes tipos de licencias y que pasamos a especificar a continuación.

En lo relativo a la captura y tratamiento de las imágenes se han utilizado licencias de *Creative Commons* [14] y *Freeware* con autorización de poder utilizarlas y editarlas con fines no comerciales. Las imágenes son tomadas de la página web de *findicons* [13].

En el desarrollo del sistema propuesto en este proyecto, para la gestión del envío de los mensajes electrónicos, se ha utilizado la librería *PHPMailer* [16]. Ésta utiliza la licencia LGPL 2.1 [15] de uso es gratuito y libre distribución.

Por último, el servidor utilizado en el proyecto, XAMPP, está bajo licencia GNU GPL [19], por lo que su uso es gratuito y de libre distribución. Sin embargo, los servicios de los cuales está compuesto, algunos utilizan otras licencias, como es el caso de *MySQL*, que tiene su propia definición de licencia de uso, se puede consultar en [17] para más detalles. Los productos de *Apache Software Foundation*, como *Apache HTTP Server* funcionan bajo la licencia de la misma organización, que se puede encontrar en [18] y el servicio *OpenSSL* funciona bajo su propia licencia también, que se puede consultar en [20]. Todos los servicios mencionados, en un principio, pueden usarse gratis siempre que se distribuya el código del proyecto.

CAPÍTULO 9: CONCLUSIONES

La culminación de este primer ciclo del proyecto nos permite presentar las siguientes conclusiones:

- Se ha diseñado y construido un prototipo de aeronave para las pruebas del sistema, concretamente un cuadricóptero.
- Se ha diseñado y adquirido los elementos esenciales de la infraestructura tecnológica para el funcionamiento básico del sistema (aeronave, *switch* inalámbrico y portátil).
- Las aeronaves pueden equiparse de diferentes tipos de módulos para adecuarse según el objetivo de la aeronave o su misión.
- Se reciben las imágenes del módulo de la videocámara.
- Se puede realizar un control sobre los accesos de los usuarios al sistema desarrollado.
- Las operaciones de las aeronaves pueden ser controladas por un usuario dado de alta en el sistema, a través de pulsaciones de teclas.
- Es posible realizar una conexión desde un navegador (probado con *Mozilla Firefox* y *Google Chrome*) de un equipo informático (un portátil) con la aeronave prototipo diseñado para este proyecto.

CAPÍTULO 10: LÍNEAS FUTURAS

Durante la realización de este primer ciclo se han ido señalando los problemas con los que se han encontrado y que no han podido ser resueltos en este ciclo ya finalizado. La solución a los mismos puede ser abordada perfectamente en las líneas futuras que aquí se proponen:

- Solucionar las diferencias de sincronización de cada ESC por software, basándose en datos que se podrán obtener realizando pruebas unitarias con cada ESC, de esta forma se obtendrá la frecuencia de funcionamiento independiente. Será necesario una herramienta para medir las revoluciones por minuto de cada motor, para controlar las velocidades de giro.
- Diseñar y mejorar las operaciones de maniobra automáticas para la aeronave, tales como el despegue y aterrizaje automático. Para ello, será necesario utilizar módulos adicionales de *Microduino*.
- Mejorar el tiempo de retardo actual (de 1 a 4 segundos) de una orden de operación, actualizando el protocolo de comunicación de TCP a UDP. Adicionalmente, se puede estudiar y mejorar el formato de los mensajes de comunicación, con la finalidad de reducir instrucciones en el algoritmo de reconocimiento de operación, lo cual debería aumentar la velocidad de reacción de la aeronave ante una orden de operación.
- Instalación de un módulo *Microduino* adicional con el objetivo de recolectar informaciones de vuelo y enviárselo al sistema para que se muestren en la interfaz web. Esto no se ha podido realizar en este ciclo, debido a que provocaba interferencias a las señales enviadas a los ESC, por lo que se genera micro acelerones en los motores.
- Implementación de la gestión de multiusuarios y multiaeronaves para SIMACA, además de la gestión de los módulos de cada aeronave.
- Diseñar e implementar programador de rutas, utilizando módulos adicionales, tales como el GPS y módulos *Microduino* adicionales.
- Diseñar e implementar módulos de seguridad pasivos y activos. Dado que será necesario revisar estos aspectos de vital importancia, para la seguridad de las personas, así como poder intentar obtener las licencias de vuelo profesional para los drones a utilizar en el sistema implementado.

BIBLIOGRAFÍA

- [1] C. Ortega Solé, Implementación de un tunnel in the sky para micro UAV, Escola d'Eninyería de Telecomunicació i Aeroespacial de Castelldefels (Universitat Politècnica de Catalunya), 30 de mayo de 2013.
- [2] X. Legasa Martín-Gil, Cuadricóptero Arduino por control remoto Android, Facultat d'Informàtica de Barcelona - Universitat Politècnica de Catalunya, 3 de Julio de 2012.
- [3] Microduino Studio, «Microduino,» [En línea]. Available: www.microduino.cc. [Último acceso: 07 04 2014].
- [4] DJI, «<http://www.dji.com>,» DJI, [En línea]. Available: <http://www.dji.com/product/phantom-2>. [Último acceso: 26 Mayo 2014].
- [5] World Wide Web Consortium, «<http://www.w3.org>,» [En línea]. Available: <http://www.w3.org/TR/html5>. [Último acceso: 26 Mayo 2014].
- [6] Parrot SA, «<http://www.parrot.com>,» [En línea]. Available: <http://ardrone2.parrot.com>. [Último acceso: 26 Mayo 2014].
- [7] World Wide Web Consortium & Internet Engineering Task Force, «<http://www.w3.org>,» [En línea]. Available: <http://www.w3.org/Protocols>. [Último acceso: 26 Mayo 2014].
- [8] S. Chamling, «<http://sanwebe.com>,» Sanwebe.com, [En línea]. Available: <http://www.sanwebe.com/2013/05/chat-using-websocket-php-socket>. [Último acceso: 26 Mayo 2014].
- [9] Shotaro...@gmail.com, «<https://code.google.com>,» [En línea]. Available: <https://code.google.com/p/jquery-websocket/>. [Último acceso: 26 Mayo 2014].
- [10] Sann-Remy, «<http://www.srchea.com>,» [En línea]. Available: <http://srchea.com/build-a-real-time-application-using-html5-websockets>. [Último acceso: 26 Mayo 2014].

- georgen...@gmail.com y roy.laur...@gmail.com, «<https://code.google.com>,» [En línea]. Available: <https://code.google.com/p/phpwebsocket/>. [Último acceso: 26 Mayo 2014].
- [12] Agencia Estatal de Seguridad Aérea, «<http://www.seguridadaerea.gob.es>,» 7 Abril 2014. [En línea]. Available: http://www.seguridadaerea.gob.es/media/4229776/el_uso_de_los_drones.pdf. [Último acceso: 27 Mayo 2014].
- [13] FindIcons.com, «<https://www.findicons.com>,» [En línea]. Available: <http://www.findicons.com>. [Último acceso: 27 Mayo 2014].
- [14] Creative Commons, «<http://creativecommons.org>,» [En línea]. Available: <http://creativecommons.org/licenses/>. [Último acceso: 27 Mayo 2014].
- [15] GNU; Free Software Foundation, «<http://www.gnu.org>,» [En línea]. Available: <http://www.gnu.org/licenses/lgpl-2.1.html>. [Último acceso: 27 Mayo 2014].
- [16] GitHub, «<https://github.com/PHPMailer/PHPMailer>,» [En línea]. Available: <https://github.com/PHPMailer/PHPMailer>. [Último acceso: 27 Mayo 2014].
- [17] Oracle, «www.mysql.com,» [En línea]. Available: <http://www.mysql.com/about/legal/licensing/oem/>. [Último acceso: 2 Junio 2014].
- [18] Apache Software Foundation, «www.apache.org,» [En línea]. Available: <http://www.apache.org/licenses/>. [Último acceso: 2 Junio 2014].
- [19] GNU & Free Software Foundation, «www.gnu.org,» [En línea]. Available: <http://www.gnu.org/licenses/gpl-2.0.html>. [Último acceso: 2 Junio 2014].
- [20] The OpenSSL Project, [En línea]. Available: <http://www.openssl.org/source/license.html>. [Último acceso: 2 Junio 2014].
- [21] International Telecommunication Union, Radio Regulation, vol. I, nº 5, 2012.

ANEXO A: MEMORIA TÉCNICA

Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino

Memoria Técnica

Autor: GenLei Cui Ji

Tutor: Víctor Nieto Lluís

Fecha: 10/02/2014

Índice

Propuesta	A-2
Descripción	A-2
Lista de componentes	A-3
Lista de herramientas	A-14
Presupuesto	A-15
Plan de ejecución	A-16
Problemas y Solución	A-21
Conclusión	A-23
Bibliografía	A-23

Propuesta

El proyecto consiste en construir un prototipo de cuadricóptero al que se le va a incorporar una unidad de control diseñada con los módulos de Microduino, plataforma de hardware libre derivada del proyecto Arduino, la que se programará para comunicarse por WiFi con un ordenador, que servirá de control remoto para el manejo de la aeronave.

Descripción

El objetivo principal del cuadricóptero es disponer de la capacidad de transportar una carga útil que permita acoplarle una variedad de módulos con la finalidad de ampliar su listado de tareas realizables. Por ejemplo módulos para tareas de vigilancia que incorporen videocámaras y/o micrófonos de diversos tipos, según las características de la tarea concreta a realizar. Otros posibles módulos podrían ser brazos robóticos o estaciones meteorológicos.

Los objetivos secundarios del cuadricóptero son la duración del vuelo, la cobertura de la señal de comunicación, retardos en la comunicación, la estabilidad de vuelo y la geolocalización de la propia aeronave.

La duración del vuelo es un factor decisivo para toda operación que se lleve a cabo, dado que fija el límite de tiempo máximo disponible para poder desplazarse hacia el destino, realizar la tarea planificada y regresar al origen. Por lo que se intentará ampliar este parámetro al máximo, pero siempre ajustándolo al presupuesto.

La cobertura de la señal de comunicación es esencial para que el operador no pierda el control de la aeronave, la señal ha de ser suficientemente fuerte, sin interferencias y con un rango admisible o en su defecto ampliable.

El retardo de la comunicación es un factor decisivo para operaciones en tiempo real, dado que puede determinar el éxito o fracaso de la misma. Éste parámetro depende mucho de la tecnología escogida para el medio de comunicación y del protocolo de comunicación que se escoja.

La estabilidad de vuelo también es un parámetro de gran importancia debido a las variadas condiciones meteorológicas al que puede estar sometida la aeronave, podría provocar accidentes graves.

En cuanto a la geolocalización es un dato esencial para localizar la aeronave y saber en todo momento con exactitud su situación para, en caso de accidente, poder localizarlo con rapidez.

Lista de componentes

Los componentes se han escogido teniendo en cuenta el presupuesto y los objetivos del proyecto.

En primer lugar, para la construcción del prototipo de cuadricóptero previsto en este proyecto se optó por buscar un diseño de chasis prefabricado que cumpliera los requerimientos de ser ligero, resistente, de bajo coste y que proporcionase espacio suficiente para montar en él todos los componentes necesarios. La decisión final recayó en el chasis HJ450 ([Figura 13](#)), por satisfacer plenamente los requisitos previamente dichos.



Figura 13: 4-Axis HJ450

En segundo lugar, se ha escogido un motor *Brushless* F4006 ([Figura 14](#)) con una relación de rpm por voltio de 950KV, valor que satisface la recomendación de que este parámetro sea inferior a 1000KV, si el objetivo perseguido es aumentar el peso de la carga útil, puesto que a menores rpm se obtiene mayor potencia aunque ello conlleva a un cierto deterioro en la maniobrabilidad. Por otro lado, siguiendo las recomendaciones del fabricante del chasis y teniendo en cuenta que este proyecto no se van a instalar módulos de grandes pesos en el prototipo, se ha decidido elegir unas hélices del tipo 10x4.5 ([Figura 15](#)). En este punto cabría especificar que si se quisiera aumentar la capacidad de carga útil del prototipo se deberían elegir unas hélices del tipo 12x4.5, tal y como lo recomienda el fabricante de los motores.

Las medidas de las hélices afectan a la capacidad de propulsión y velocidad de giro. Estos 2 factores vienen determinados según el diámetro y el paso, siendo el diámetro el primer número (10) que mide la distancia existente de punta a punta de las palas del hélice y el segundo número es el paso (4.5) que corresponde al ángulo que forma entre el perfil de las palas respecto del eje horizontal de giro. Cuanto mayor sea el diámetro, mayor será la capacidad de propulsión, dado que mueve una cantidad mayor de aire, por otro lado, cuanto mayor sea el paso, menor será la velocidad de giro, puesto que se encuentra con mayor rozamiento de aire.

Otro factor importante a destacar en este apartado son los sentidos del giro de cada motor, afectando así a los diseños de las hélices. Como se puede observar en la [Figura 16](#), los motores 1 y 3 giran hacia la derecha, por lo que las hélices instaladas sobre ellos han de estar adaptadas para dicho sentido de giro, lo mismo ocurre para los motores 2 y 4, pero para girando hacia la izquierda. Esto permitirá al cuadricóptero girar durante el vuelo en el sentido que se desee. Para girar hacia la izquierda se tiene que aumentar la velocidad del giro de los motores 1 y 3, para el lado contrario hay que aumentar la velocidad del giro de los motores 2 y 4.



Figura 14: Motor Brushless Outrunner F4006



Figura 15: Hélices 10x4.5

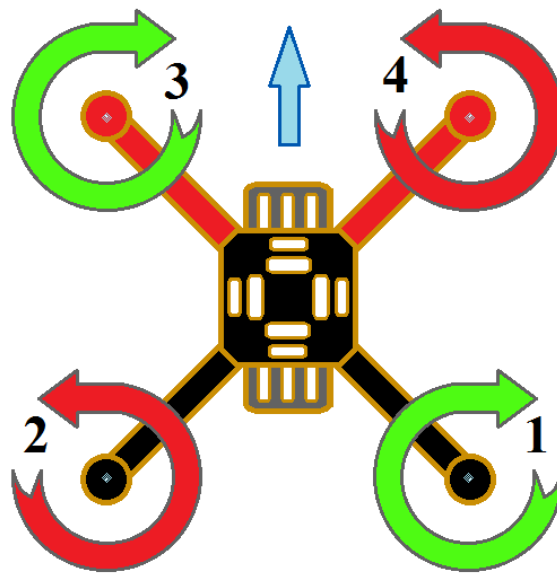


Figura 16: Sentido del giro de cada motor

En tercer lugar, se ha seleccionado una batería LiPo de 3 celdas con una capacidad de 5.200mAh ([Figura 17](#)), para alimentar a todo el sistema.

En este punto se ha de tener en cuenta que a mayor capacidad de batería, mayor es su peso, lo cual también aumenta el consumo de los motores, por lo que se ha de buscar un equilibrio entre el consumo y la capacidad, pero éste factor sólo se puede calcular una vez que toda la aeronave se haya construido para comprobar su peso neto.

Adicionalmente cabe destacar que las baterías con capacidades mayores al escogido, aumenta drásticamente el precio. Otro factor importante a considerar en las baterías es la tasa de descarga máxima soportada, en otras palabras, es la velocidad con la que se puede descargar una batería y se mide en unidades C (carga total dividida por hora). En nuestro caso la batería seleccionada tiene una tasa de descarga 30C, por lo que la intensidad máxima que puede proporcionar la batería al cuadricóptero será de

$30 \cdot (5.200\text{mAh})/h = 156.000\text{mA}$. Esta intensidad de corriente se podría mantener sólo durante $5.200\text{mAh}/156.000\text{mA} = 0,0333h \approx 2$ minutos.

Según las características de los motores escogidos, consumen como máximo 32A por motor, lo cual daría una autonomía estimada de la batería de $5200\text{mAh}/(32\text{A} \cdot 4) = 0,0406h \Rightarrow$ alrededor de 2,44 minutos. Pero éste cálculo no ha tenido en cuenta el factor del peso del aparato ni las condiciones de vuelo, por lo que es bastante impreciso.



Figura 17: Batería LiPo 3S 11,1V 5200mAh

En cuarto lugar, se ha optado por una unidad controladora de velocidad electrónica que utiliza la tecnología de *Switching* para variar la tensión de 11,1V a 5V, siendo ésta más eficiente por utilizar transistores y otros componentes electrónicos para cambiar rápidamente los voltajes logrando una menor pérdida de energía, que los transformadores lineales que utilizan resistencias [1]. El modelo escogido es el ESC (Siglas en inglés, *Electronic Speed Controller*) Thunder-40A M-SW (Figura 18), dado que su parámetro de corriente máxima debe ser mayor que la que tiene el motor que va a controlar, siendo esta de 32A.. Adicionalmente, este modelo es programable, por lo que se puede configurar algunas opciones como el tiempo de respuesta de los motores, comportamientos de apagado por bajo nivel de batería o el control del sentido de rotación de los motores, entre otras.



Figura 18: ESC Thunder 40A M-SW BEC 5V/3A

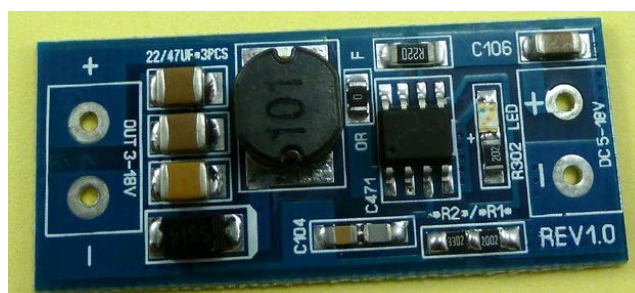


Figura 19: Transformador de 5V a 3,3V

En quinto lugar, y respondiendo a criterios de seguridad, se han adquirido una serie de componentes adicionales para poder prevenir los accidentes. Uno de ellos ha sido la adquisición de una bolsa de protección ([Figura 20](#)) para la batería, dado que ha habido casos en las que las baterías LiPo eran defectuosas y se han incendiado solas y, a veces, han provocado explosiones. Esta bolsa tiene finalidad de evitar que se propague el fuego en caso de incendiarse o resistir las ondas de expansión en caso de explosión.

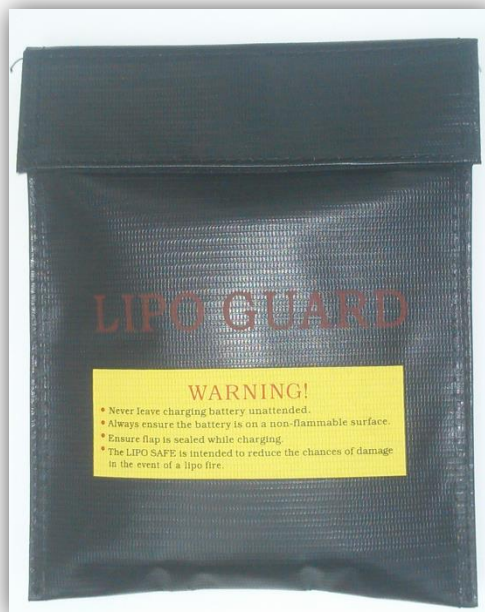


Figura 20: Bolsa de protección de baterías LiPo

Se ha decidido también adquirir una alarma ([Figura 21](#)) para avisar un nivel bajo de la batería, con el fin de prolongar la vida útil de la misma, puesto que si se llega a consumir toda la carga, según las especificaciones del fabricante, pueden provocar una reducción drástica de la vida útil de la misma.



Figura 21: Alarma Batería baja con iluminación led y zumbido

Otro componente adicional que se ha adquirido es un cargador ([Figura 22](#)) especialmente diseñado para recargar las baterías LiPo. Tan negativo resulta una descarga completa de la batería como la sobrecarga de la misma o no asegurar la misma carga en todas las celdas que la componen, lo cual podría causar graves daños a la batería



Figura 22: Cargador de baterías LiPo de 2S a 3S

Por último, se han adquirido también unos sensores ultrasónicos ([Figura 23](#)) como sensores de proximidad para controlar la distancia respecto al suelo en los momentos más decisivos, que son el despegue y el aterrizaje. El principal motivo es porque el barómetro que se incluye en uno de los módulos de Microduino, mide la altura absoluta con referencia al nivel del mar, no tiene en cuenta los posibles relieves de la superficie real.



Figura 23: Sensor ultrasónico US-020

En sexto lugar, se tomaron las decisiones pertinentes en relación al cableado a utilizar para la interconexión de los diferentes componentes electrónicos. Para la conexión de la alimentación se escogieron cables de grado 16AWG (estándar correspondiente a *American Wire Gauges*) que, aunque presentan una sección por debajo de lo recomendado para soportar 40A a 90°C, las condiciones reales de operación se mantendrán siempre por debajo de esos límites. Además es del mismo tipo del que viene instalado de fábrica en el controlador electrónico de velocidad (ESC).

En cuanto al cableado de comunicación de los componentes electrónicos, se van a utilizar cables rígidos tipo hilo para las conexiones con los módulos Microduino. Además, se usarán los cables tipo *DuPont Breadboard Jumper* ([Figura 24](#)) para conectar los sensores ultrasónicos.



Figura 24: Cables DuPont Breadboard Jumper Macho-Hembra

Se ha tomado la decisión de utilizar tubos termo-retráctiles ([Figura 25](#)) para cubrir las uniones soldadas para aportar mayor seguridad a dichas uniones, dado que por vibración o movimiento en los puntos de unión hechos con estaño, pueden aparecer grietas o hasta incluso llegar la rotura de la unión.



Figura 25: Tubos termo-retráctiles

También se ha determinado los tipos de conectores que se van a utilizar en los diferentes puntos de conexión, como por ejemplo el XT60 ([Figura 26](#)) para los cableados de suministro eléctrico y el conector T para conectarlo directamente con la batería; Los conectores tipo *Bullet* ([Figura 27](#)) para las conexiones entre el motor y el ESC.



Figura 26: Conector tipo XT60



Figura 27: Conector tipo Bullet

En octavo lugar, se ha decidido incorporar un módulo de cámara, para retransmitir imágenes en directo desde la aeronave. Ésta incorpora su propio módulo WiFi para la comunicación con el router, las imágenes son accesibles vía web *streaming*. El modelo escogido es Ai-Ball, como se puede observar en la (Figura 28), es bastante pequeño y ligero, pero se ha optado por aligerar más la cámara quitando todo elemento innecesario, como se puede observar en la Figura 33. Por otro lado, se le proporcionará la alimentación desde un transformador de 5V a 3,3V (Figura 19), éste utiliza también la tecnología *Switching* y estará conectado al *Thunder 40A M-SW* (Figura 18)



Figura 28: Cámara WiFi Ai-Ball

En noveno lugar, se toma la decisión de que la unidad controladora sea Microduino, que en comparación con el Arduino es mucho más pequeño y modular, como se puede observar en la Figura 29. Utilizando el módulo Microduino-Core+ como procesador central, que se le ampliará funcionalidades complementándolo con los módulos WiFi (Microduino-CC3000), GPS (Microduino-NEO6M) y sensores de aceleración, barómetro y brújula (Microduino-10DOF). Para poder programar la unidad es necesario disponer del módulo Microduino-FT232R, que corresponde al puerto de comunicación USB. Todos los módulos (Figura 30) serán instalados sobre la plataforma Microduino-VDUO, que permite rebajar la altura del conjunto, dispersar las conexiones y asentar los módulos con mayor firmeza, dado que tiene un pre-diseño para la fijación.

En este punto cabe destacar que varios módulos Microduino funcionan con 3,3V, por lo que se ha adquirido otro transformador (Figura 19) de 5V a 3,3V para suministrarles el voltaje adecuado.

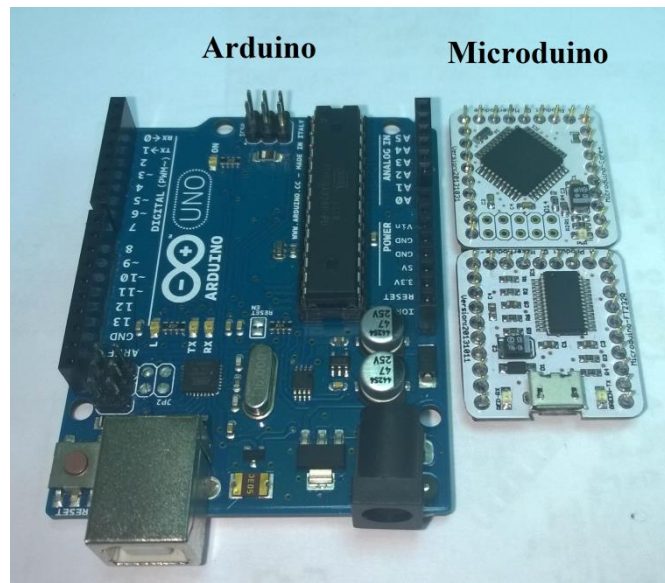


Figura 29: Diferencias de tamaño entre Arduino y Microduino

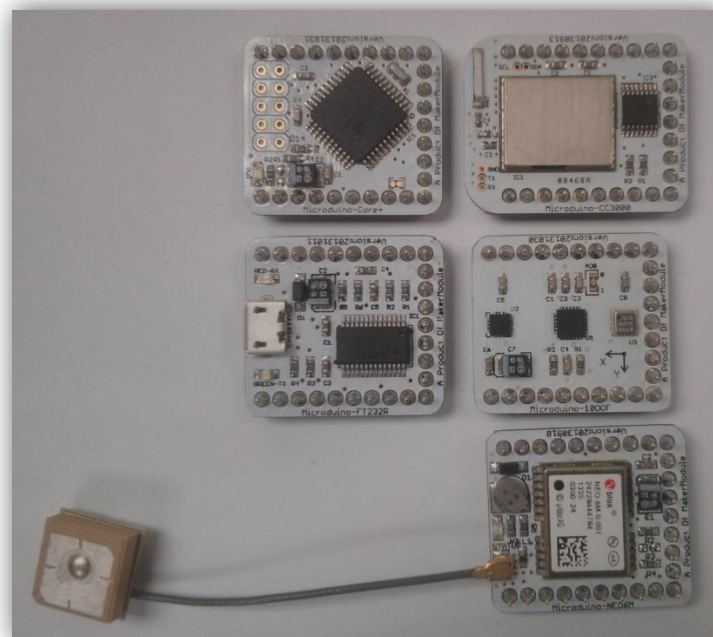


Figura 30: Módulos Microduino

En décimo y último lugar, se ha decidido utilizar bridas para las sujeciones de los diferentes elementos sobre el chasis del cuadricóptero.

Lista de herramientas

El proyecto exige la realización de soldaduras en diferentes puntos, como las cabezas de los conectores, las uniones de los cables y quizás de algunos elementos electrónicos pequeños, por lo que se necesitará un soldador potente y con punta fina, además se precisa de pasta de soldar, para ayudar las posibles soldaduras con circuitos electrónicos y disponer también de una esponja especial para limpiar la punta del soldador, en resumen, es necesario disponer de una estación de soldadura semi-profesional.

La mayor parte de las soldaduras requeridas se realizan en cableados y conectores, por lo que se hará uso de los tubos termo-retráctiles para sellarlos mejor y ayudar a reducir movimientos en los puntos de unión. Para ello es necesario utilizar algún tipo de fuente de calor para aplicarlo sobre los tubos, basta con utilizar un mechero o en su defecto, utilizar la punta del soldador.

Durante las soldaduras, algunos componentes se pueden calentar mucho, por lo que es necesario disponer de una mesa habilitada para este tipo de trabajo y además disponer de un o un sistema similar para sujetar los componentes a soldar y así evitar accidentes.

En el momento de atar las bridas para las sujeciones de los cableados y otros elementos, es necesario disponer de unas tijeras para cortar las partes sobrantes. También son necesarias para cortar los cables a la longitud requerida y pelar sus extremos a soldar.

En la parte de montaje de los motores, es imprescindible disponer de un juego de destornilladores para poder apretar los tornillos, que pueden tener cabezas tipo H (hexagonales) y/o Cruces. Se recomiendan disponer al menos de H2.0 y H2.5.

Por último, es necesario disponer de un cable micro-USB para cargar los programas en el Microduino-Core+.

Presupuesto

Componente	Precio/unidad	Unidades	Total
Microduino-Core+	10\$	1	10\$
Microduino-FT232R	12\$	1	12\$
Microduino-CC3000	30\$	1	30\$
Microduino-10DOF	24\$	1	24\$
Microduino-NEO-6M	30\$	1	30\$
Microduino-Vduo	8\$	1	8\$
Heat shrink tube 20mm	2.5\$	1	2.5\$
ESC Thunder 40ª M-SW	15.7\$	4	62,8\$
Cable DuPont Breadboard (pack 40 piezas)	4.5\$	1	4.5\$
Conector tipo T	2.3\$	1	2.3\$
Cargador de batería LiPo	10.99\$	1	10.99\$
Conector tipo Bullet (pack 20 pares)	7.99\$	1	7.99\$
Hélices verdes 10x4.5R (pack 2 unidades)	3.1\$	2	6.2\$
Hélices naranjas 10x4.5R (pack 2 unidades)	3.1\$	1	3.1\$
Motor F4006 KV950 (pack 4 unidades)	51.8\$	1	51.8\$
Batería LiPo 5200mAh	32\$	1	32\$
Conector tipo XT60 (pack 8 pares)	9.7\$	1	9.7\$
Chasis 4-Axis HJ450	18.4\$	1	18.4\$
Alarma batería baja	3.46\$	1	3.46\$
Bolsa de protección para batería	7\$	1	7\$
Sensor ultrasónico US-020	4.39\$	3	13.17\$
Transformador reductor 5V->3V3 (pack de 2)	3.84\$	1	3.84\$
Cámara Ai-Ball	71.40\$	1	71.40\$
Cableado 16AWG (10 pies)	6.8\$	1	6.8\$
Router neutro WiFi TP-Link TL-WR841N	36.80\$	1	36.80\$
Kit tubos termo-retráctiles	14.85\$	1	14.85\$
Total		32	483.6\$

Tabla 1: Presupuesto

En total se ha invertido 483.6\$, que son aproximadamente 354,70€ a día de 10/02/2014.

Cabe destacar que en el presupuesto no se incluyen las herramientas.

Plan de ejecución

Primero, se procedió a realizar el montaje del chasis, uniendo las diferentes piezas con tornillos tipo H2.0, como se puede observar en la [Figura 31](#):



Figura 31: Chasis del cuadricóptero montado

Segundo, se realizó las conexiones eléctricas provisionales para comprobar el correcto funcionamiento de los diferentes componentes electrónicos recibidos. Para verificar el funcionamiento de los sensores ultrasónicos, se cargó un programa ejemplo de la librería *NewPing* en el Microduino-Core+ para contrastar los resultados obtenidos de ellos con una regla milimétrica. Las comprobaciones de los motores y de los ESC se realizaron utilizando el módulo Microduino-Core+ junto a un programa de test utilizando test utilizando la librería *Servo* para verificar la correcta sincronización de los ESC en el control de las revoluciones del motor. Para comprobar los transformadores se les conectó a una fuente de alimentación de 5V y se verificó que el voltaje de salida correspondía a los 3,3V esperados. La comprobación de la cámara Ai-Ball, se realizó utilizando una pila CR2 de 3V para su alimentación, se le conectó a un router externo y se verificó la correcta transmisión de vídeo en web streaming. Finalmente, para comprobar los módulos Microduino, se han utilizado algunos programas facilitados por el fabricante como de ejemplo y otros facilitados por las librerías oficiales de Arduino.

Tercero, en esta fase se procedió a soldar los diferentes tipos de conectores a los componentes y cables correspondientes. A los cables de los motores se les soldó el conector macho del tipo *Bullet*, como se puede observar en la [Figura 31](#). A los cables del ESC hacia los motores se les soldaron con los conectores hembras del tipo *Bullet* y a los cables que van hacia la batería se les soldó el conector macho del tipo XT60, como se puede apreciar en la [Figura 32](#). Los cables del grado 16AWG, utilizados para interconectar la batería con los ESC, por un extremo se les soldaron los conectores hembra del tipo XT60, para conectarlos a los ESC y por el otro extremo se soldaron todos unidos a los dos polos correspondientes del conector tipo T, como se puede observar en la [Figura 32](#), para conectarlo a la batería. La cámara Ai-Ball, sin la carcasa externa, se le soldó 2 cables rígidos tipo hilo para conectarlo al transformador reductor, tal como se muestra en la [Figura 33](#).

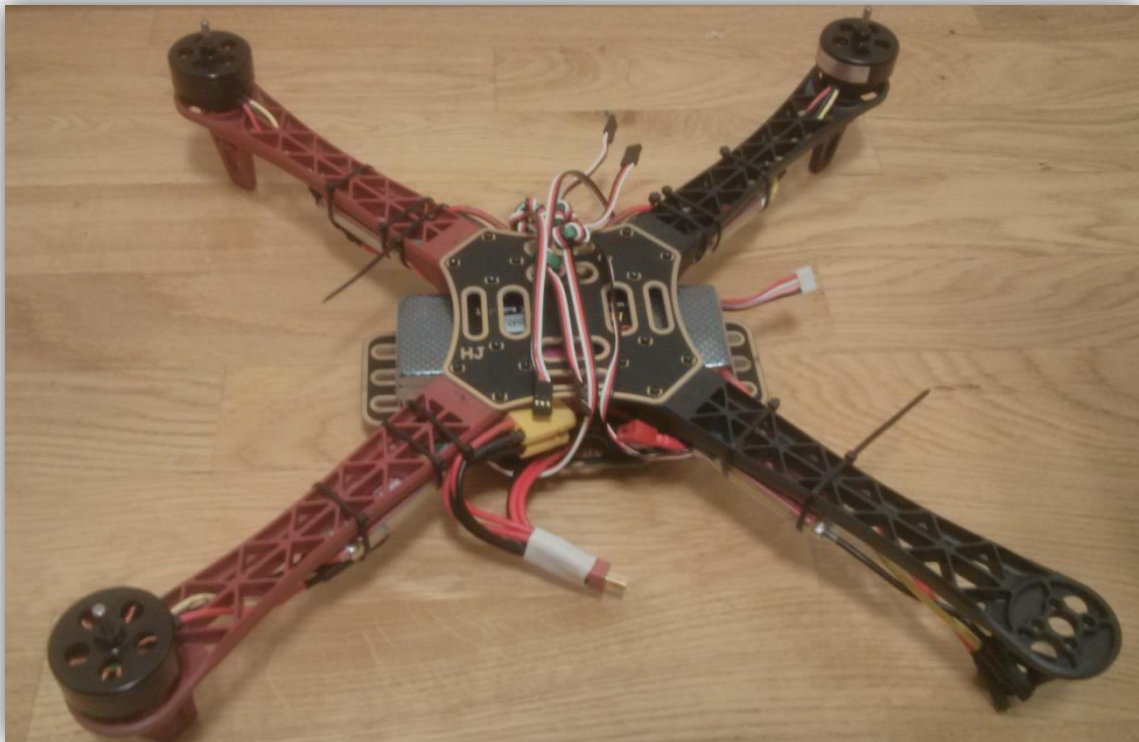


Figura 32: Cuadricóptero con los ESC y motores montados en el chasis

Cuarto, tras comprobar el correcto funcionamiento de todos los componentes y que los conectores estén soldados, se procedió a acoplar los diferentes componentes al chasis. Primeramente, se montaron los ESC y los motores en cada brazo del chasis, situando a los primeros por debajo de los brazos del chasis y utilizando bridas para sujetarlas. Los motores se colocaron en los sitios previstos para ello en cada brazo del chasis y se fijaron utilizando tornillos tipo H2.5.

Quinto, se ajustó las longitudes de los cables de control de los ESC, pero comenzando con las de alimentación de 5V (cable rojo), a dos de las cuatro se les soldó correspondientemente a una transformadora de 5V a 3,3V, colocadas en los brazos del chasis del color rojo, como se puede observar en la [Figura 34](#). El tercer cable rojo de los ESC se les ha soldado a los cables para alimentar los dos sensores ultrasónicos colocados en los brazos del chasis del color negro. El cuarto cable rojo de los ESC se le conectó directamente al pin de 5V de las placas de Microduino.

Acto seguido, los cables de la polaridad negativa (color negro) de los ESC, han sido conmutadas para que los ESC puedan funcionar debidamente. Posteriormente, se han conectado los cables de control (color blanco) de los ESC en los pines correspondientes de la placa Microduino-Core+. Estos últimos cables son blandos y está formado por varios hilos de cobre, por lo se les han soldado el último tramo con un cable rígido del tipo hilo para facilitar la conexión con los pines de la placa Microduino-Core+.

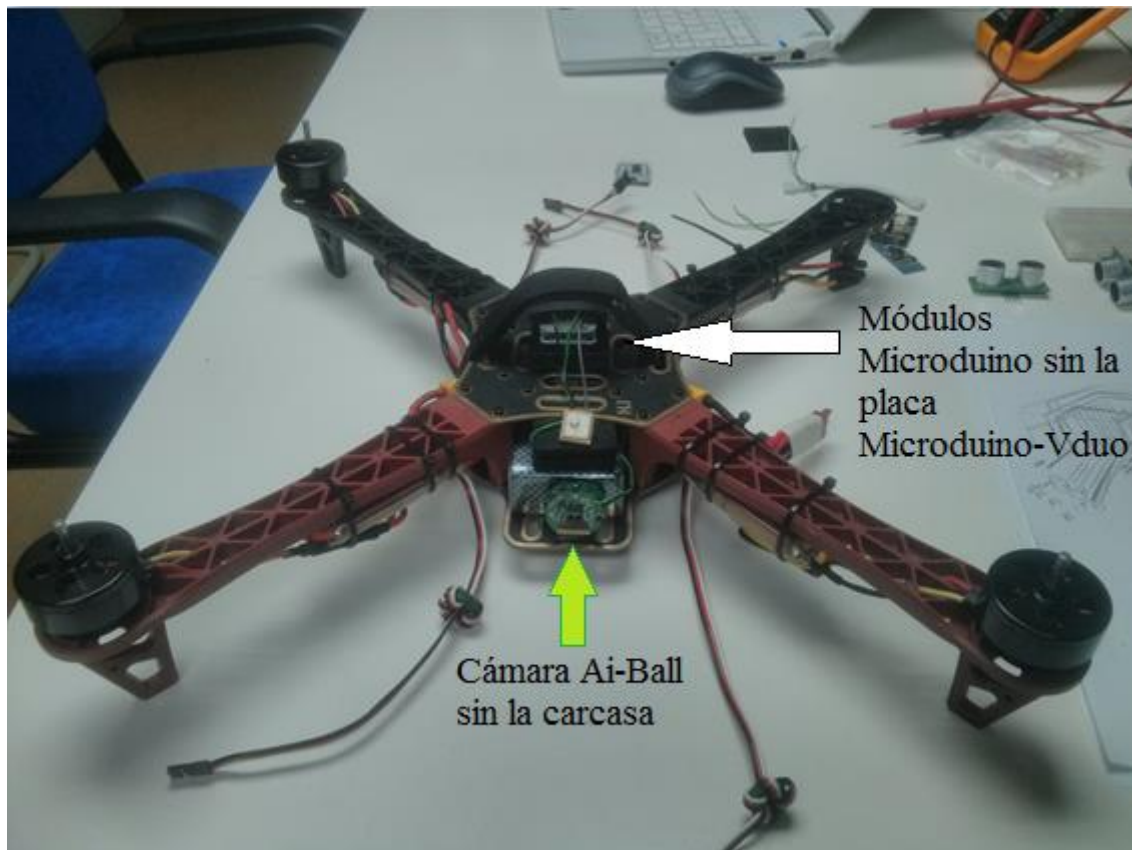


Figura 33: Cuadricóptero con los módulos Microduino instalados

Sexto y último punto, Se fijó el módulo Microduino-Vduo a la placa superior del chasis del cuadricóptero, mediante bridas. Durante esta fase, las conexiones de los diferentes cables se conectaron a los pines correspondientes del módulo Microduino-Core+ a través del Microduino-Vduo, como se puede observar en la [Figura 34](#) y la [Figura 35](#).

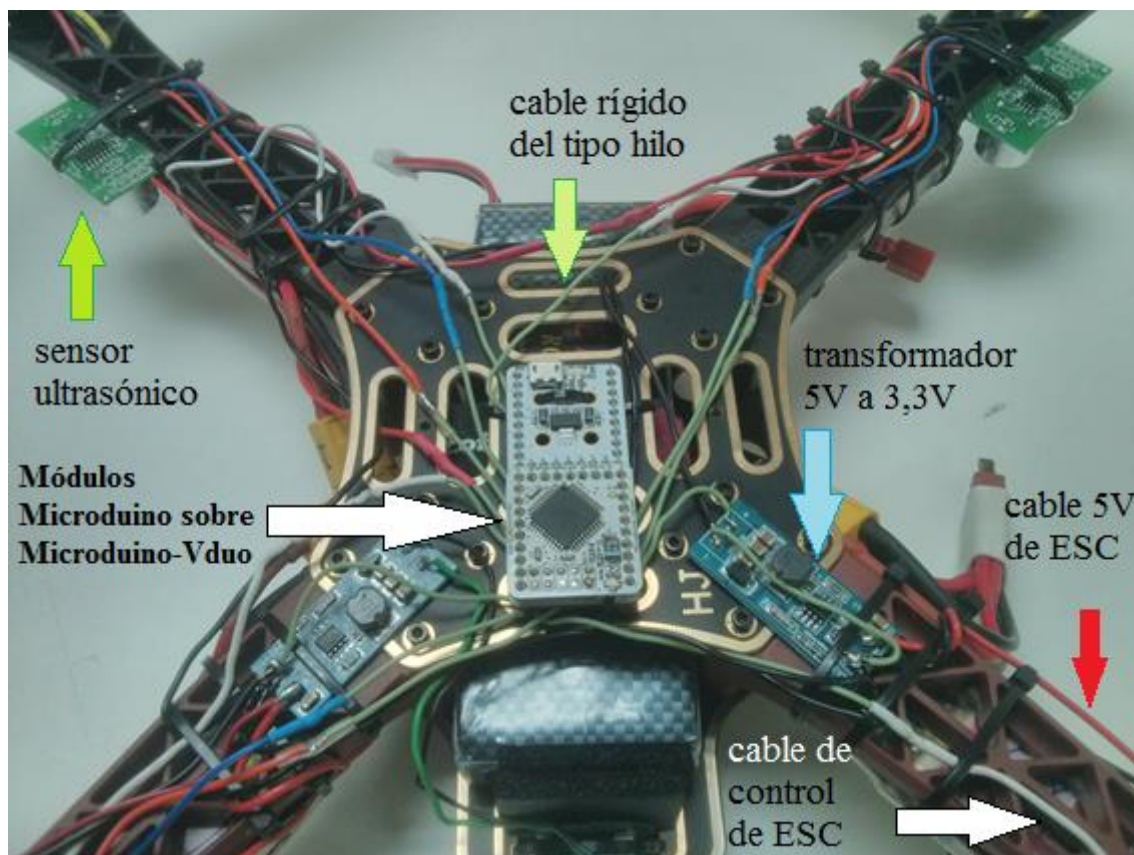


Figura 34: Cuadricóptero con el módulo Microduino-Vduo y Microduino-Core+ instalados

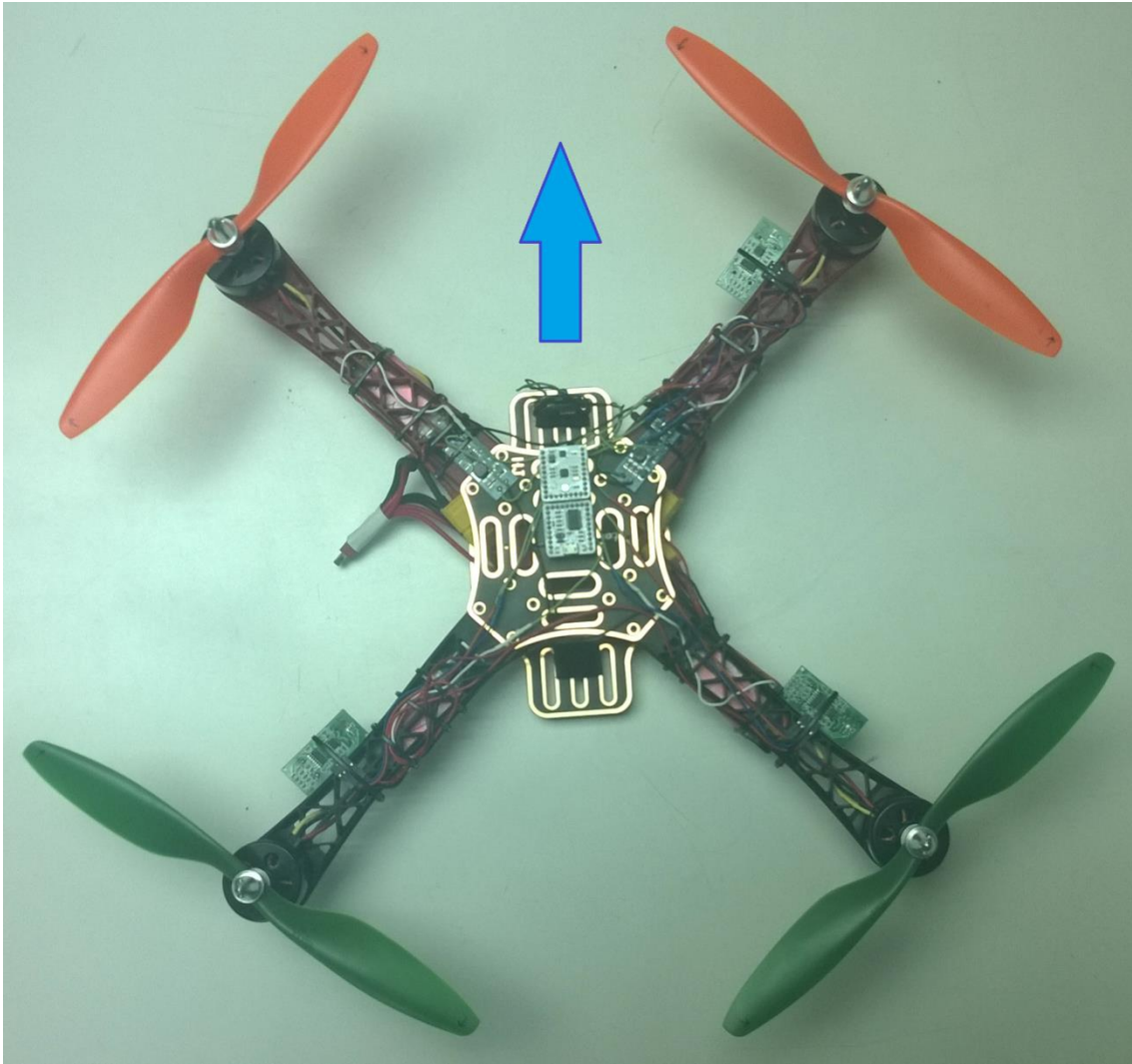


Figura 35: Cuadricóptero con las hélices montadas

Problemas y Solución

Se presentaron diversos problemas con el suministrador que incidieron negativamente en los tiempos de inicio de este proyecto, tales como falta de material suficiente en sus almacenes, retrasos en los envíos y no disponibilidad de algunos tipos de elementos de fijación, como por ejemplo los tornillos H2.5, teniendo que reemplazarlos por tornillos de otro tipo.

En el momento de las conexiones de los cables de control de los ESC a los pines de los módulos de Microduino, se encontró con el problema de que no se podían fijar adecuadamente los módulos con el chasis del cuadricóptero, lo que provocaba que a veces se salieran algunos cables de los pines, debido a los movimientos. Esto se solucionó incorporando al diseño el módulo Microduino-Vduo, dado que dispone de varios agujeros donde poder pasar bridas, lo que permite sujetarlos firmemente al chasis y evitar una desconexión accidental de los cables, permitiendo además reducir la altura del apilamiento de los módulos Microduino, tal y como se puede comparar entre la [Figura 33](#) y la [Figura 34](#).

La colocación de la batería en la placaba inferior del chasis también ha supuesto un reto, debido a que el propio kit del chasis incluía una tira de velcro para sujetar la batería, pero sólo aportaba sujeción lateral. Esto no permitía garantizar que la batería no pudiera desplazarse durante el vuelo a lo largo de su eje longitudinal, poniendo con ello en grave riesgo la seguridad de la operación con esta aeronave. La solución que se encontró es que el propio kit incluía también una tira de velcro que es adhesiva por la parte de atrás, por lo que se decidió recortar una porción de 2cm x 2,15cm, para que no sea muy difícil a la hora de desacoplarlo, ni muy fácil de soltarse.

Otro problema que se tuvo, es con la localización de la cámara, que está justo a un lado de la batería. El problema reside en que la cámara puede sobrecalentarse bastante, ello podría afectar al rendimiento o vida útil de la batería, por lo que se optó por una solución provisional de colocar 2 esponjas de 4mm de grosor cada una, como aislante. No obstante, se quiere hacer constar que durante el funcionamiento de la aeronave, las hélices generan una fuerte corriente de aire y con ello podría disipar parte del calor desprendido por los diferentes componentes electrónicos, por lo que en un principio, no debería de ser un gran problema.

A la hora de instalar los módulos Microduino, se presentó un problema de conectividad, dado que el módulo Microduino-Core+ dispone de 10 pines adicionales, pero que no están soldados por defecto, sin embargo, por razones de diseño, estos pines deben soldarse y ello dificulta y complica la conexión de los cables y el módulo Microduino-Core+. La solución adoptada es la siguiente, debido a que el módulo Microduino-Vduo viene prediseñado con unos agujeros justo debajo de un lateral, coincidiendo exactamente con los huecos de los pines de expansión del Microduino-Core+, se ha decidido soldar los conectores de los pines extras en el módulo Microduino-Vduo y a la vez éstos soldarlos con el módulo Microduino-Core+, que estará justo encima, tal como se observa en la [Figura 36](#):

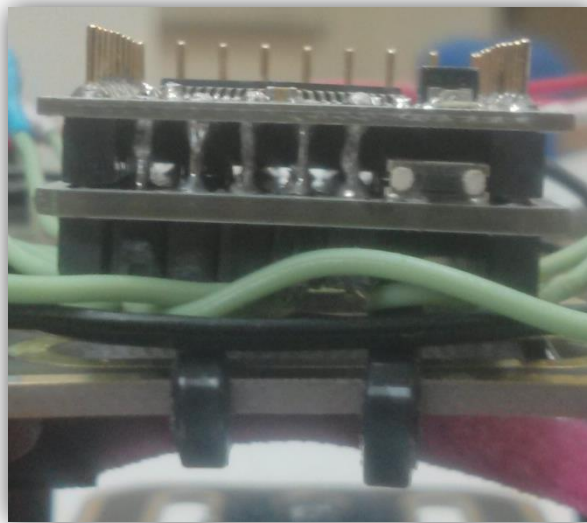


Figura 36: Soldadura entre Microduino-Core+ (Encima) y Microduino-Vduo (Debajo)

Por último y el problema más importante, es la de garantizar la seguridad durante la operación con el cuadricóptero, dado que las hélices están al aire libre y durante el funcionamiento podría rozar con obstáculos, que podrían ser personas y causarle daños físicos. Para resolver este problema existen soluciones constructivas de carácter pasivo tales como barreras de protección a las hélices mediante rejillas de materiales resistentes y muy ligeros pero, sin duda, la medida más sencilla y segura es la de no permitir el uso de este tipo de aeronaves en lugares con riesgo físico para las personas. Adicionalmente, conviene comentar que también existen medidas de carácter activo, como sensores de proximidad, que podrían instalarse para desviar la aeronave en caso de posible peligro de choque con objetos en la dirección de su trayectoria o prever acciones de parada de los motores en caso de haber sufrido un accidente.

Conclusión

La construcción del cuadricóptero ha supuesto bastante trabajo, sobretodo en la fase inicial, la parte de tomar las decisiones de qué componentes adquirir e instalar en la aeronave. Puesto que no es una competencia propia de la carrera de Ingeniería Informática. Este proyecto ha aportado finalmente gran cantidad de nueva información acerca de componentes electrónicos, aeromodelismo, soldadura electrónica y me ha permitido, a su vez, aprender cosas nuevas relativas a la carrera, como es el caso del entorno de desarrollo del Arduino, un modelo de desarrollo, un poco peculiar, aunque no extraño ni nuevo, pero supone una adaptación a la hora de desarrollar algún sistema con ello.

El desarrollo de esta parte del proyecto ha sido bastante rápido en lo que se refiere a la construcción de la aeronave, pero aún queda mucho por aprender más adelante y desarrollar, como por ejemplo el control y las maniobras del cuadricóptero y su comunicación con el ordenador.

Bibliografía

- [1] On Semiconductor, "Linear & Switching Voltage Regulator Handbook," in *HB206/D*, 2002, pp. 69-70.

ANEXO B: ESPECIFICACIÓN DE REQUISITOS DEL SISTEMA

Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino

Especificación de Requisitos del Sistema

Autor: GenLei Cui Ji

Tutor: Víctor Nieto Lluís

Fecha: 24/02/2014

Índice

1. Introducción	B-3
2. Ámbito y alcance	B-3
3. Glosario	B-4
4. Requisitos del sistema	B-5
Requisitos iniciales	B-5
Requisitos no funcionales	B-5
Features	B-6
Feature 1: Gestión de usuarios	B-6
Feature 2: Operaciones de maniobra	B-8
Feature 3: Gestión de aeronaves	B-9
Feature 4: Gestión de módulos o perfil de aeronaves	B-9
Bibliografía	B-11

1. Introducción

En el presente documento se detalla la Especificación de Requisitos de Software (ERS) del sistema “Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino”, en adelante SIMACA. El SIMACA consistente en una interfaz web, un servidor de comunicaciones y la controladora de la aeronave. Se explicarán la finalidad, las funcionalidades, las interfaces, las restricciones y la respuesta ante estímulos exteriores de cada componente del sistema.

El documento está realizado siguiendo las directrices de [1], motivo por el cual se definen una serie de terminologías en la sección [3 Glosario](#), para evitar ambigüedades.

Adicionalmente, en la sección [2 Ámbito y alcance](#), se explicará de manera global y resumida el funcionamiento del sistema. En la sección [4 Requisitos del sistema](#), se explicarán todos los requisitos del sistema, que adicionalmente estarán identificados numéricamente y clasificados por features, según la técnica basada en la filosofía de las necesidades del usuario son satisfechas por una serie de *features* del sistema a desarrollar, basado en la pirámide *NEEDS-FEATURES-REQUIREMENTS* [2].

2. Ámbito y alcance

El SIMACA consiste en crear un sistema con el cual un usuario utilizando una interfaz web desde un ordenador pueda manejar a distancia una aeronave a través del WiFi con el teclado o ratón.

El usuario podrá interactuar directamente con una interfaz de página web para realizar todas las operaciones de maniobra de la aeronave. Dichas operaciones incluyen, ascender, descender, desplazarse hacia delante (avanzar), hacia atrás (retroceder), hacia la derecha o izquierda y girar hacia la derecha o izquierda. Además el usuario tendrá a su disposición una interfaz web que le permitirá agregar una nueva aeronave al sistema o eliminar una existe y otra interfaz para editar el perfil de una aeronave que esté dada de alta en el sistema, pudiendo activar o desactivar los diferentes módulos instalados en la aeronave.

Las operaciones de maniobra por parte del usuario son enviadas directamente por WiFi al ordenador de a bordo.

Las aeronaves y los módulos integrados en ellas no son competencia directa de SIMACA, debe existir otra entidad que se encargue de ello, pero el sistema sí que facilita una herramienta para integrar los datos de las nuevas aeronaves a SIMACA. Dicha herramienta se explicará en la sección [4 Requisitos del sistema](#).

3. Glosario

Aeronave: Se define como todo vehículo que es capaz de volar, pero en el proyecto sólo se contemplan los no tripulados y que se clasifican como de aeromodelismo.

Cuadricóptero: Es un tipo de aeronave que utiliza cuatro motores eléctricos, cada uno situado en una de las cuatro alas del chasis principal que conforman una cruz, como sistema de propulsión y control de direcciones de la misma.

Features: Son funcionalidades de cualquier sistema que se fundamentan en las necesidades reales del usuario final. Tienen la característica de utilizar el lenguaje del usuario final, en vez de ser más técnico.

Herramienta de integración de los módulos: Es una función implementada con el lenguaje PHP, con el cual se reconoce todos los datos incluidos en el documento XML facilitado, para su posterior incorporación en la base de datos.

Interfaz web: Es la parte visual y representativa de las *features* definidas en la sección 4, requisitos del sistema. El usuario podrá satisfacer todas sus necesidades interactuando con ella, bien mediante pulsaciones de teclas o haciendo clics con el ratón. En ella pueden aparecer contenidos de uno o varios módulos instalados en la aeronave.

Módulo: Es un conjunto de componentes electrónicos que conforman una unidad electrónica y cumple una finalidad determinada, cuyo funcionamiento es totalmente independiente de otras posibles unidades, a excepción de la alimentación eléctrica.

Operación de maniobra: Es una orden o una serie de órdenes de movimiento a la aeronave, que implica un desplazamiento de la misma desde una posición inicial a la dirección y/o sentido requerido. Adicionalmente, puede acompañarse de peticiones a los diferentes módulos instalados en ella.

Perfil de la aeronave: Es una interfaz web desde donde el usuario final puede observar y modificar las características de los módulos instalados en la aeronave.

4. Requisitos del sistema

Las *features* definidas para el sistema se van a caracterizar por los diferentes niveles o grados que tengan los parámetros que sirven para compararse, siendo estas: el Estado, la Prioridad, el Esfuerzo y el Riesgo.

Así pues, el Estado puede ser en revisión o aprobada; La Prioridad puede ser 1, 2 o 3, siendo 1 la más prioritaria; El Esfuerzo puede ser bajo, medio o alto; El Riesgo puede ser bajo, medio o alto

Requisitos iniciales

SIMACA necesita en primer lugar, colaborar con una entidad adicional que le facilite los documentos XML, donde se reflejan todas las características de cada aeronave entregada y los módulos integrados en ella.

En segundo lugar, necesita un entorno donde pueda trabajar con el protocolo TCP/IP para comunicarse con las aeronaves. Al mismo tiempo, las aeronaves deben disponer de módulos inalámbricos para poder comunicarse con SIMACA, el proyecto se centrará en utilizar la tecnología WiFi 802.11b/g, pero dicho elemento queda sujeto a las decisiones de la entidad mencionada anteriormente, bien pudiendo adaptarse a otras tecnologías o versiones, siempre que sea compatible con el protocolo TCP/IP.

En tercer lugar, necesita estar comunicada con una infraestructura de redes inalámbricas con una cobertura tan extensa como así lo requieran las operaciones. Esta cobertura de redes inalámbricas será ofrecida por la misma entidad adicional mencionada con anterioridad y así como del mantenimiento.

Requisitos no funcionales

SIMACA tendrá un portal de acceso restringido solamente al personal autorizado.

Las comunicaciones entre el SIMACA y la máquina del usuario final estarán cifradas utilizando el cifrado SSL.

Una aeronave sólo puede ser manejada en un instante por un único usuario. Por tanto, si existe un usuario en la interfaz de manejo de una aeronave determinada, ningún otro usuario, ni administrador, podrá acceder a dicha interfaz.

Features

Feature 1: Gestión de usuarios

Estado: En revisión

Prioridad: 1

Esfuerzo: Alto

Riesgo: Alto

1. El acceso a SIMACA está restringido al personal autorizado, para utilizar los servicios se ha de registrar previamente en el sistema de alta de usuarios.
2. Los usuarios del sistema pueden realizar las funcionalidades que estén descritas en las [Feature 2](#): Operaciones de maniobra y [Feature 4](#): Gestión de módulos o perfil de aeronaves.
3. Los administradores del sistema además de disponer lo anterior, pueden realizar las siguientes funciones: Revisar y aceptar las solicitudes de alta de nuevos usuarios; Modificar el perfil de los usuarios; Eliminar usuarios del sistema y las funcionalidades descritas en la [Feature 3](#): Gestión de aeronaves. Los administradores son usuarios predefinidos en el sistema.
4. El formulario de alta de usuario ha de disponer de los siguientes campos: nombre y apellidos reales del usuario, DNI, dirección de correo electrónico, un teléfono de contacto y una fotografía del usuario. El sistema solicitará al usuario repetir el campo de correo electrónico en un campo adicional, con la finalidad de verificar que se ha introducido correctamente.
5. Una vez enviado el formulario, se le notificará mediante un mensaje en la interfaz web que la solicitud se ha enviado correctamente y debe esperar una respuesta de los administradores, que le llegará en su correo electrónico. En caso de error se le notificará un mensaje indicando el error producido.
6. Los administradores dispondrán de una interfaz web donde podrán gestionar las nuevas solicitudes de alta y otra para eliminar los usuarios del sistema.
7. En la interfaz de revisión de solicitudes de alta, los administradores pueden visualizar un listado de solicitudes, acompañado de los datos enviados por el usuario, con la finalidad de verificar que todos los datos están correctos y en tal caso, podrán aceptar la solicitud y el sistema automáticamente generará un correo electrónico y se lo enviará con un enlace al usuario, para confirmarle que se le ha aceptado la solicitud y que tiene que crear una cuenta escogiendo un nombre de usuario y la contraseña. En caso contrario, se le enviará también un correo para avisar al usuario de tal decisión.
8. Los administradores podrán realizar cualquier cambio en los perfiles de los usuarios, seleccionándolo de una lista, que puede ser filtrado por nombres de los usuarios, pero los usuarios sólo podrán acceder a su propio perfil y las únicas acciones permitidas son la de cambiar la contraseña de la cuenta y visualizar sus datos.

9. Para aquellos usuarios que hayan olvidado la contraseña de su cuenta, existirá un formulario para que puedan solicitar la asignación de una nueva contraseña, el sistema enviará la respuesta de la petición al correo con el que se registró el usuario.
10. Las contraseñas y los nombres de las fotografías recogidas en los formularios de solicitud y registro estarán guardadas y cifradas en la base de datos.

Feature 2: Operaciones de maniobra

Estado: En revisión

Prioridad: 1

Esfuerzo: Alto

Riesgo: Alto

11. El usuario podrá seleccionar de un listado la aeronave que desea manejar, pudiendo filtrar el listado por el parámetro de nombre de aeronave.
12. Todas las operaciones de maniobra posibles dependen del tipo de aeronave. Estas operaciones están recogidas en el documento XML asociado a la aeronave en cuestión. El mismo documento es facilitado por la entidad responsable de suministrar unidades físicas de aeronaves para ser manejados con SIMACA.
13. El sistema utilizará de referencia un prototipo de aeronave cuadricóptero, cuyas operaciones disponibles son: avanzar, retroceder, desplazar hacia izquierda o hacia derecha, girar hacia la derecha o la izquierda, despegar, aterrizar, elevar y descender.
14. El usuario podrá utilizar las teclas del ordenador para manejar la aeronave. Para ello se ha asignado la siguiente configuración por defecto para el prototipo de cuadricóptero:
 - Tecla 'W' para avanzar
 - Tecla 'A' para desplazar hacia izquierda
 - Tecla 'S' para retroceder
 - Tecla 'D' para desplazar hacia derecha
 - Tecla 'H' para descender
 - Tecla 'U' para elevar
 - Tecla 'I' para girar hacia izquierda
 - Tecla 'L' para girar hacia la derecha
 - Tecla '1' para despegar
 - Tecla '0' para aterrizar.
 - Tecla 'Enter' para parada de emergencia
15. No obstante, el usuario podrá personalizar la combinación de teclas desde el perfil de la aeronave e igualmente podrá manejar la aeronave utilizando el ratón, para ello la interfaz web ofrecerá botones para las operaciones descritas anteriormente.
16. La interfaz web ofrecerá estas operaciones independientemente de otros módulos instalados en la aeronave.
17. Las órdenes del usuario final serán transmitidas en tiempo real hacia la aeronave.
18. Las informaciones generadas en las aeronaves serán recogidas por el sistema y mostrada en la interfaz del usuario en tiempo real.

Feature 3: Gestión de aeronaves

Estado: En revisión

Prioridad: 2

Esfuerzo: Alto

Riesgo: Bajo

19. En esta interfaz web, los administradores podrán acceder a otra para añadir una nueva aeronave (siendo esta la herramienta de integración de aeronaves) importando los datos desde un fichero XML. También podrán acceder otra interfaz web para eliminar las aeronaves de SIMACA.
20. Los documentos XML, se presupone que ya están bien formados y validados con un Schema XML, a implementar con el sistema.
21. En la interfaz web, el usuario puede acceder a otra interfaz web donde podrá consultar y visualizar un listado de aeronaves instaladas en SIMACA, cuyos resultados aparecen en forma de una tabla, en la que una columna será una imagen de la aeronave, en otra aparecerá el nombre de cada aeronave y ambos serán un acceso directo hacia el perfil de la aeronave. Por último, habrá una última columna donde aparecerá un botón para eliminarlos, que será identificado como un icono representativo junto la palabra de “eliminar”.
22. En el proceso de eliminación de una aeronave, el sistema solicitará confirmación preguntando al usuario final de si realmente desea eliminarlo.
23. En la interfaz del listado de las aeronaves, los administradores pueden filtrar los resultados mediante parámetros como: nombre de aeronave o nombre de módulos.
24. Todas las aeronaves tendrán un nombre representativo y único, facilitando su identificación para el usuario final.

Feature 4: Gestión de módulos o perfil de aeronaves

Estado: En revisión

Prioridad: 3

Esfuerzo: Alto

Riesgo: Medio

25. El usuario escoge de un listado, el perfil de la aeronave al que desea acceder. Esta lista se puede filtrar por el nombre de aeronave.
26. En la interfaz web se mostrará informaciones de la propia aeronave como una imagen representativa o descriptiva de la misma, el nombre de la aeronave, datos técnicos como la capacidad de carga útil, el tiempo de vuelo, número de módulos instalados, versión de la controladora instalada, el peso neto, número de propulsores, capacidad de la batería y las medidas físicas del chasis, además de los módulos instalados.

27. En esta interfaz web el usuario podrá visualizar todos los módulos instalados en la aeronave en cuestión. Adicionalmente, tendrá la opción de editarla, pulsando el botón de editar perfil, situado en la parte superior de la página web. Así como podrá agregar módulos pulsando al icono representativo junto a la frase “agregar nuevo módulo” situado debajo del último módulo mostrado en la interfaz web. También podrá eliminar un módulo instalado pulsando la opción de “eliminar módulo” localizado dentro del recuadro de cada módulo que aparece en el perfil.
28. Además, algunos módulos instalados podrán recogerá la telemetría de la aeronave, datos como la altura absoluta, altura relativa, velocidad, localización, dirección o sentido magnético (brújula), inclinación, tiempo de vuelo, capacidad de batería restante, etc. Dichos datos serán recogidos con cierta frecuencia a determinar por cada módulo. Cada petición será realizada por SIMACA directamente hacia la aeronave.

Bibliografia

- [1] *IEEE Recommended Practice for Software Requirements Specifications*, 1998.
- [2] D. Leffingwell y D. Widrig, *Managing Software Requirements: A Use Case Approach*, 2nd ed. Addison-Wesley, 2003.

ANEXO C: DISEÑO DE ALTO NIVEL

Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino

Diseño de Alto Nivel

Autor: GenLei Cui Ji

Tutor: Víctor Nieto Lluís

Fecha: 28/02/2014

Índice

<u>Introducción</u>	C-2
<u>Modelo Ambiental</u>	C-3
<u>Descripción del sistema</u>	C-3
<u>Diagrama contexto</u>	C-3
<u>Lista de acontecimientos</u>	C-3
<u>Matriz de trazabilidad</u>	C-12
<u>Modelo de Comportamiento</u>	C-14
<u>DFD</u>	C-14
<u>Nivel 0</u>	C-14
<u>Nivel 1</u>	C-15
<u>Nivel 2</u>	C-16
<u>Nivel 3</u>	C-17
<u>Diagrama de entidad-relación</u>	C-18
<u>DD</u>	C-19
<u>Unidad controladora del cuadricóptero</u>	C-21
<u>Protocolo de comunicación</u>	C-23

Introducción

El presente documento tiene la finalidad de describir un diseño de alto nivel para el sistema que se quiere implementar.

A continuación se exponen las consideraciones adoptadas al abordar este diseño:

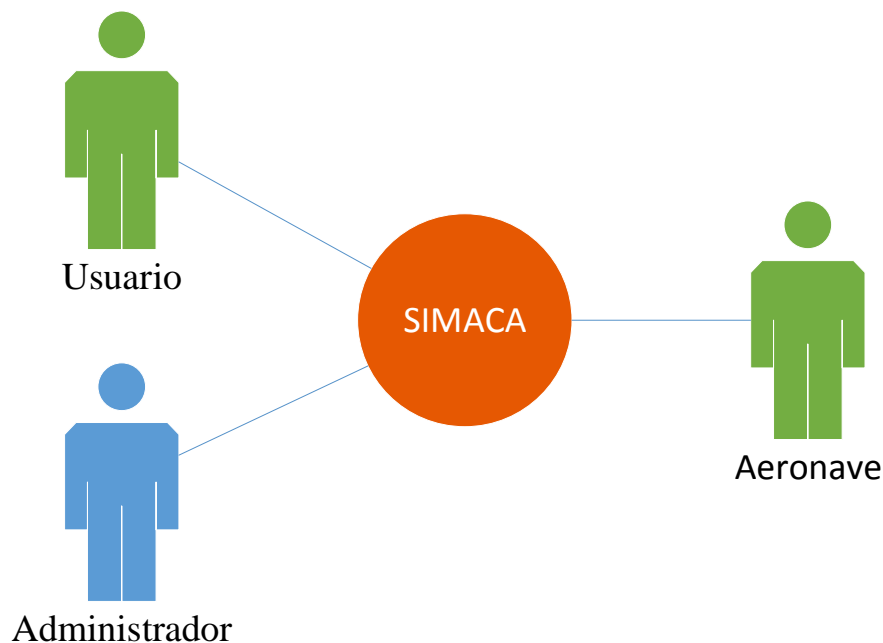
- En los acontecimientos AC21 y AC22, se entiende que cualquier listado de resultados de usuarios y/o aeronaves pueden ser filtrados por el usuario o el administrador, según el caso, utilizando los parámetros predefinidos.
- Se entiende que la aeronave es una entidad porque es capaz de recibir información del sistema, interpretarla e incluso generar una respuesta al sistema o generar una nueva información de entrada para el sistema.
- Se ha decidido no reflejar en el documento los casos en el que la aeronave genera información de entrada para el sistema. Las informaciones que puede generar la aeronave son aquellas que son definidas en el documento XML, utilizado para dar de alta la aeronave. Puesto que la información puede variar de una aeronave a otra, se ha decidido solucionar este problema en el diseño de bajo nivel, para casos concretos, centrándose en la unidad de aeronave prototipo.

Modelo Ambiental

Descripción del sistema

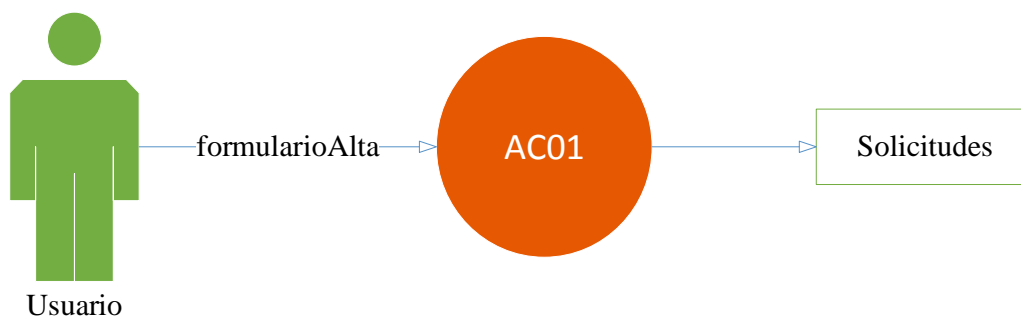
El sistema a implementar es el proyecto “Diseño de un sistema para el manejo a distancia de aeronaves contruidos con arduino”, que de ahora en adelante será conocido como SIMACA. El SIMACA está formado por una interfaz web, un servidor de comunicaciones y la controladora de la aeronave. El usuario final podrá manejar una aeronave utilizando la interfaz web, con la que puede realizar acciones como emitir órdenes de movimientos hacia la aeronave, dar de alta o de baja a las aeronaves y modificar los perfiles de cada aeronave agregando o suprimiendo módulos en ella.

Diagrama contexto

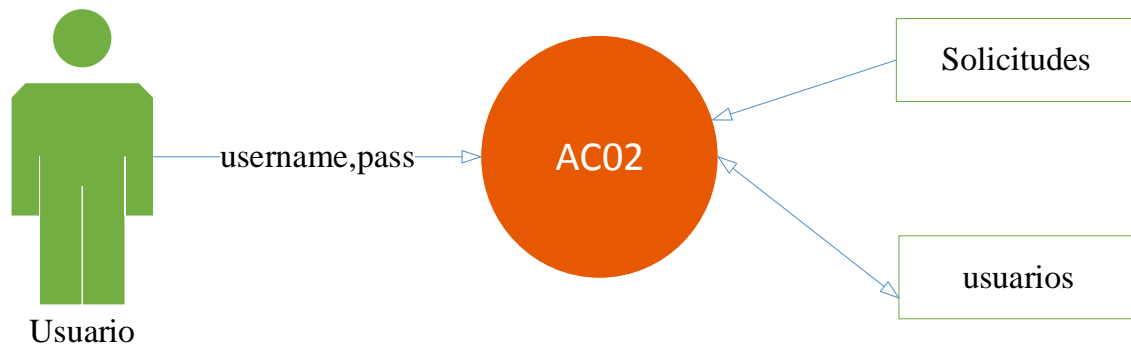


Lista de acontecimientos

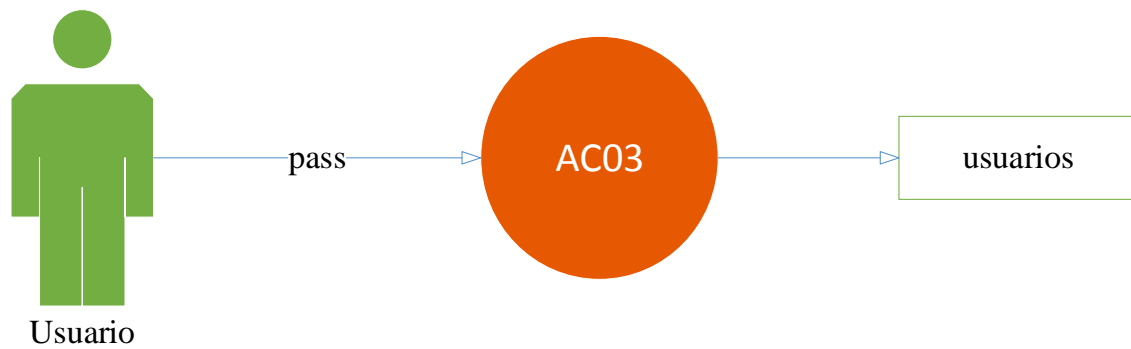
AC01: El usuario envía el formulario de alta en el sistema.



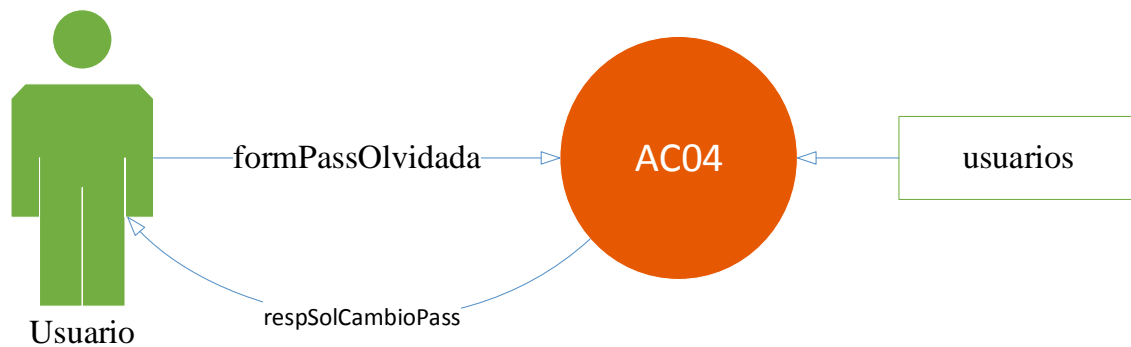
AC02: El usuario crea una contraseña y nombre de usuario para su nueva cuenta.



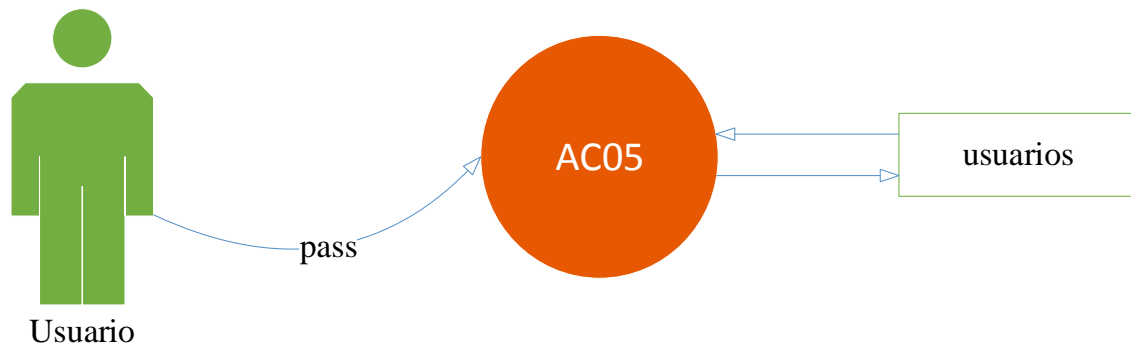
AC03: El usuario cambia la contraseña de su cuenta desde su perfil.



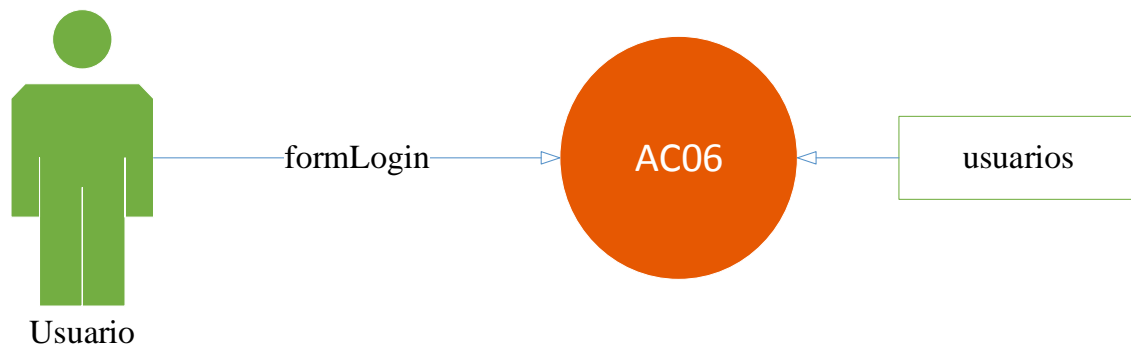
AC04: El usuario solicita reasignar la contraseña de su cuenta por olvido.



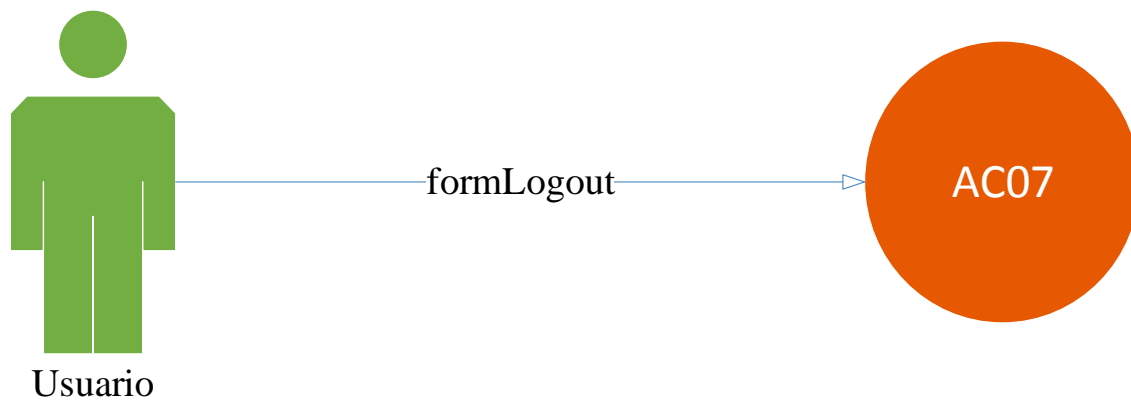
AC05: El usuario asigna la nueva contraseña de su cuenta por olvido.



AC06: El usuario se autentica en el sistema.



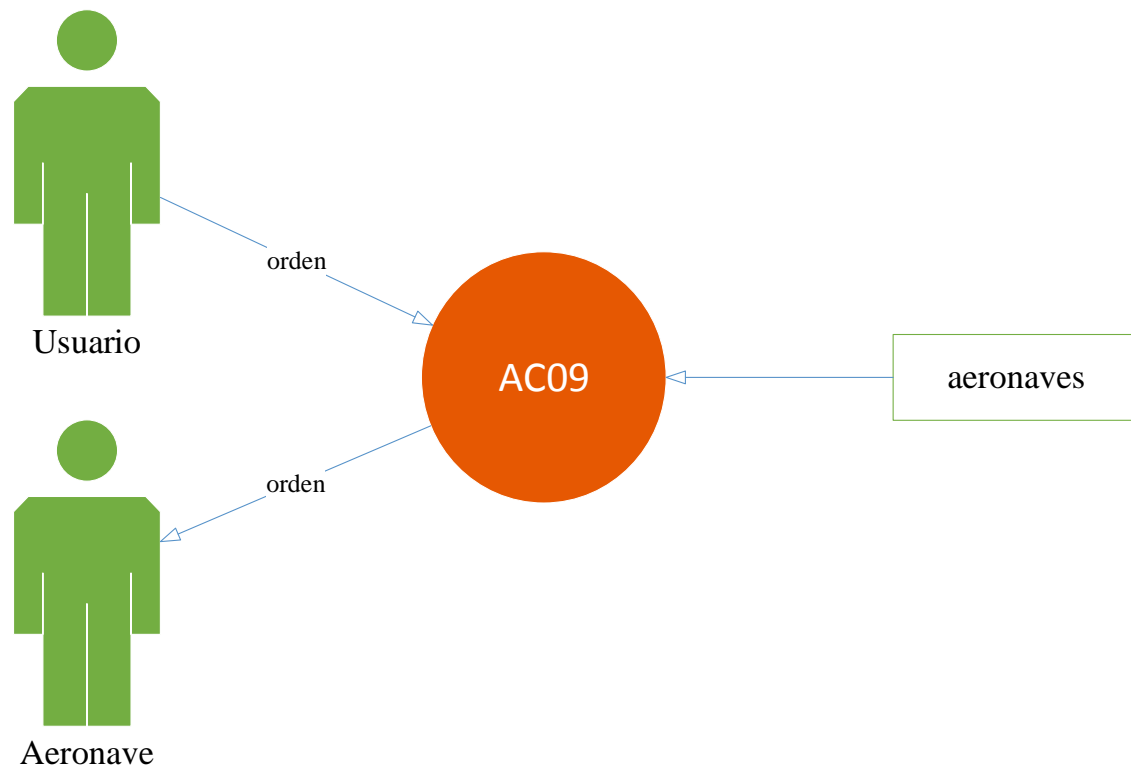
AC07: El usuario cierra su sesión en el sistema.



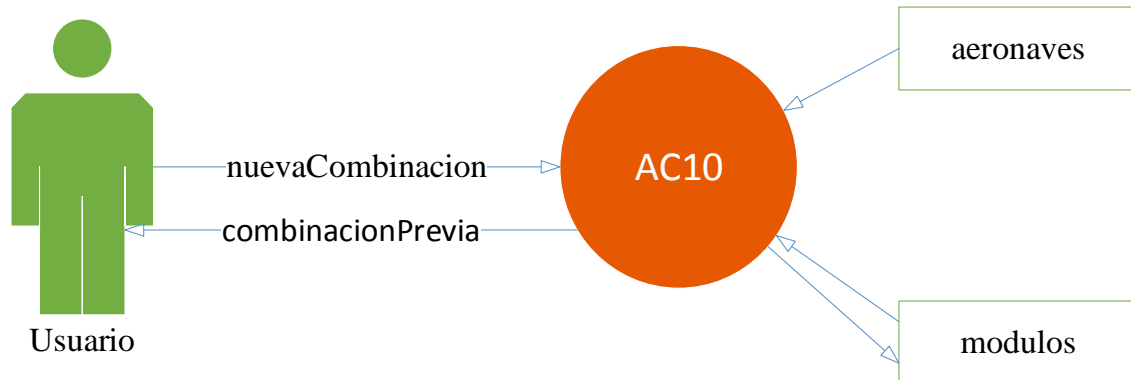
AC08: El usuario selecciona una aeronave para manejarla.



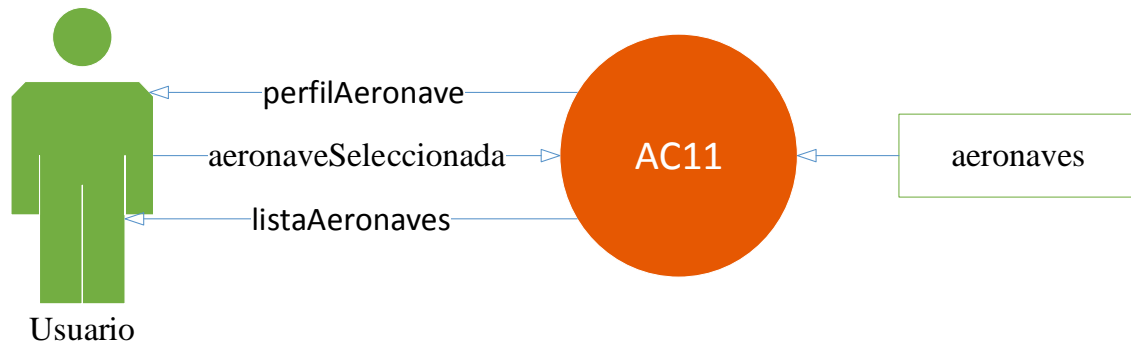
AC09: El usuario envía órdenes de maniobras a la aeronave.



AC10: El usuario personaliza la combinación de teclados para el manejo de la aeronave desde la interfaz de manejo.



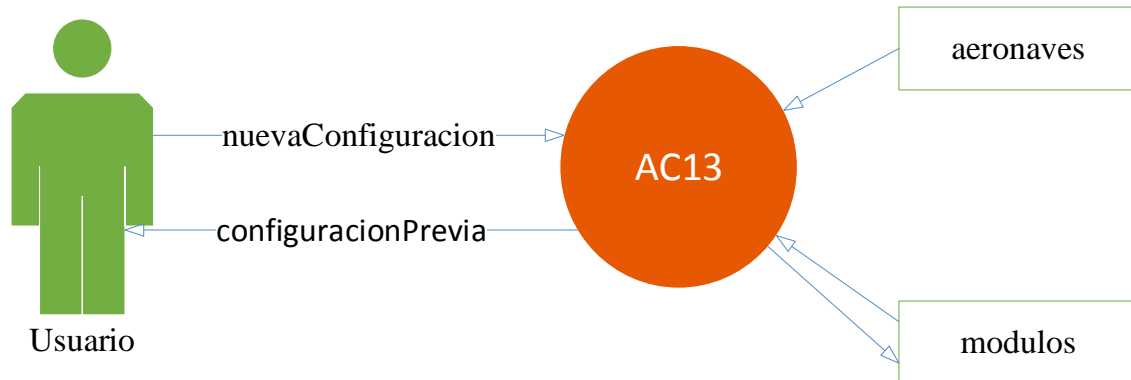
AC11: El usuario selecciona el perfil de una aeronave.



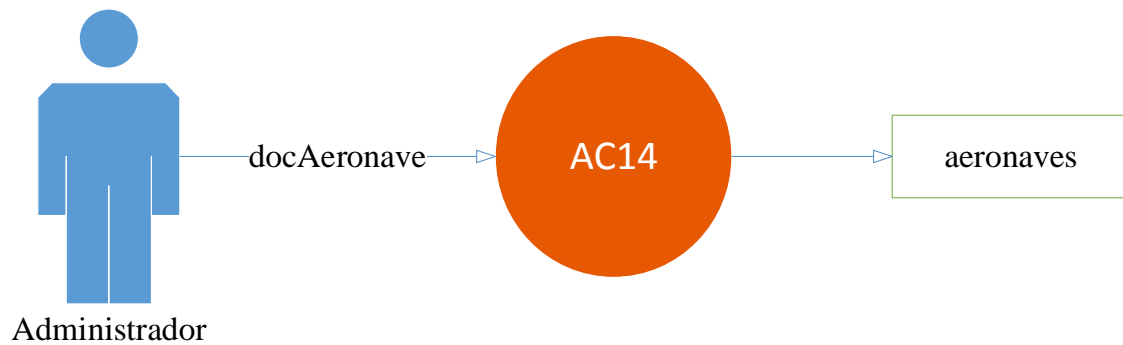
AC12: El usuario edita el perfil de una aeronave.



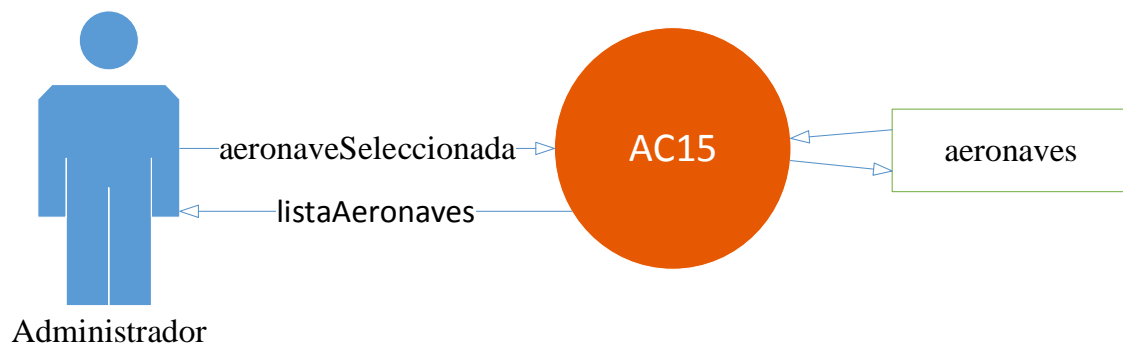
AC13: El usuario modifica las configuraciones de un módulo determinado desde el perfil de la aeronave.



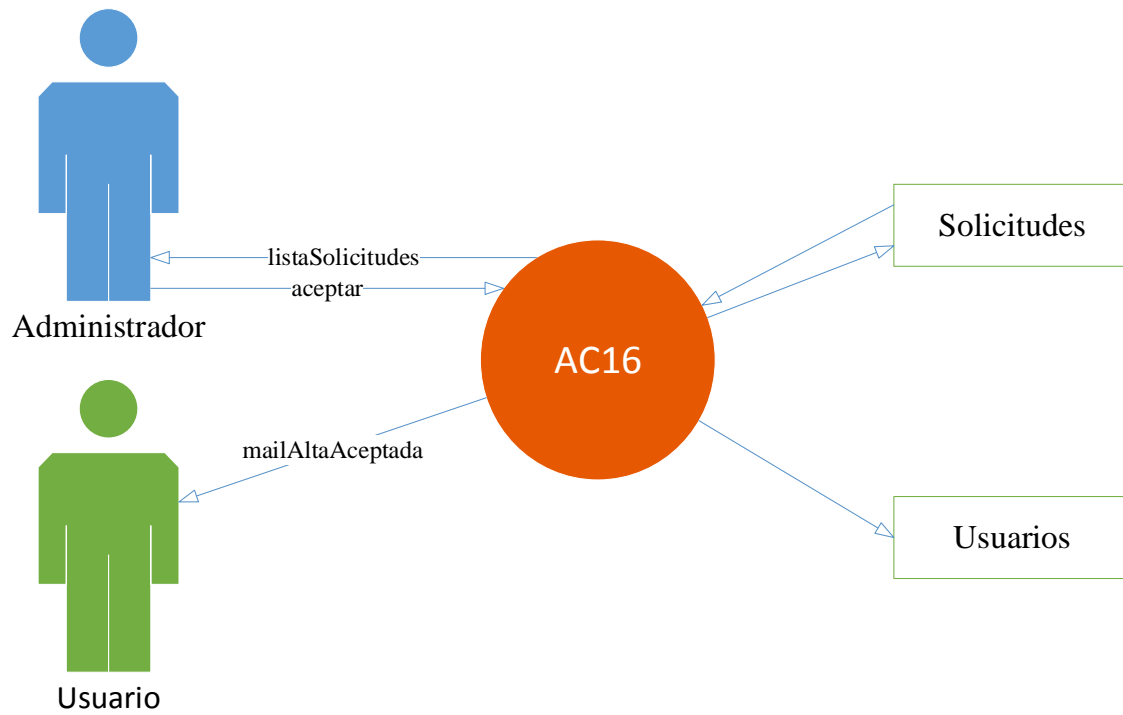
AC14: El administrador añade una aeronave al sistema.



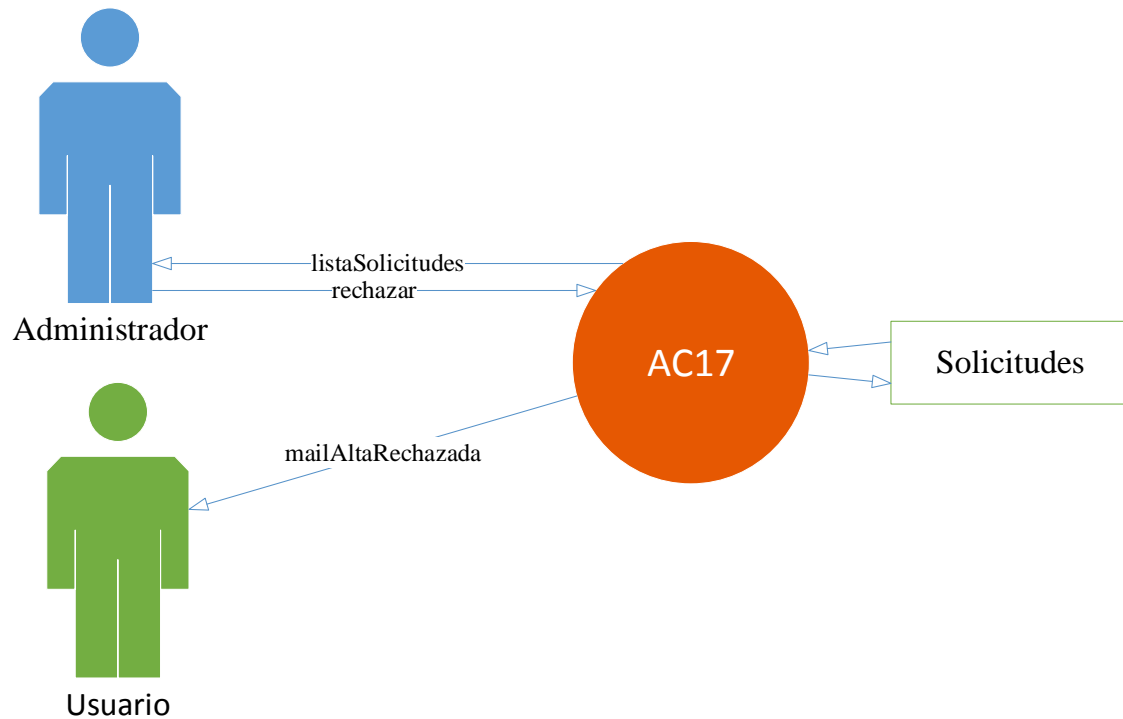
AC15: El administrador elimina una aeronave del sistema.



AC16: El administrador revisa y acepta la solicitud de un alta.



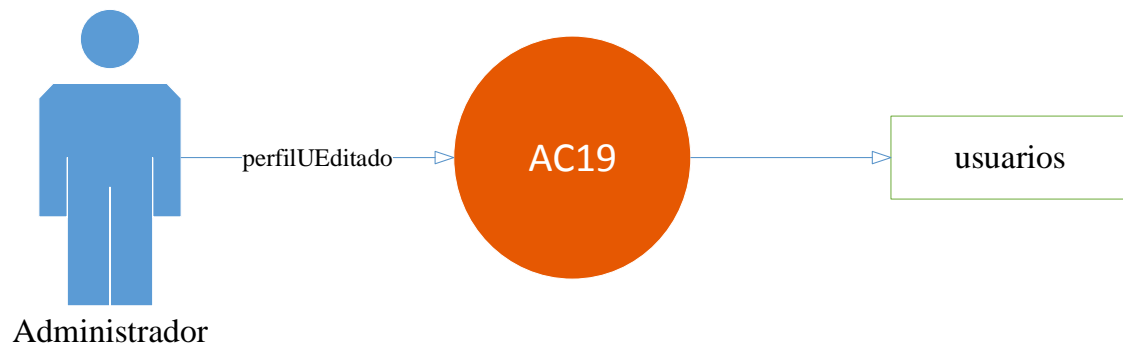
AC17: El administrador revisa y rechaza la solicitud de un alta.



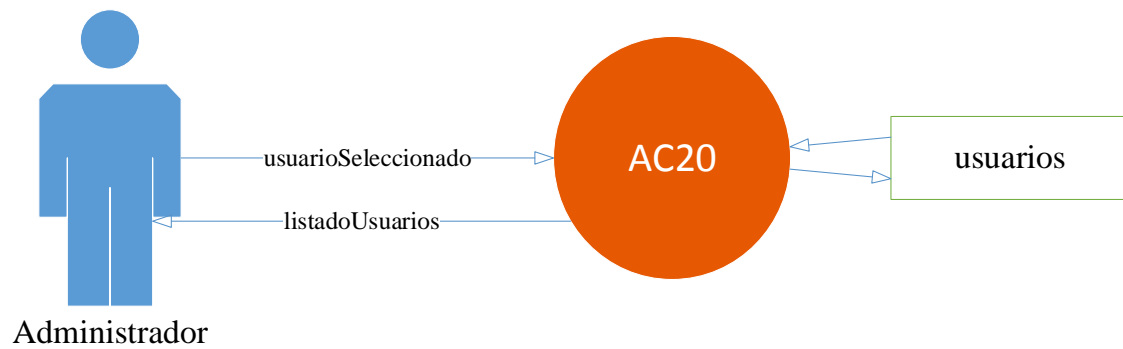
AC18: El administrador selecciona el perfil de un usuario a modificar.



AC19: El administrador edita el perfil de un usuario.



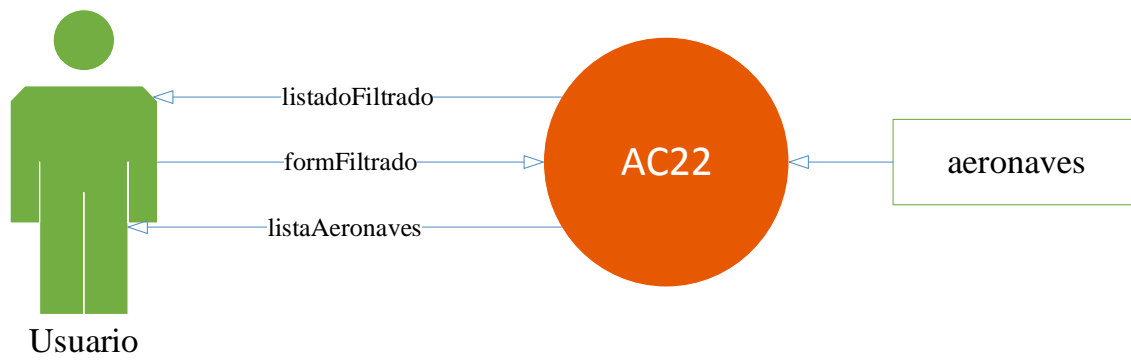
AC20: El administrador elimina un usuario del sistema.



AC21: El administrador filtra los resultados de un listado de usuarios.



AC22: El usuario filtra los resultados de un listado de aeronaves.



Matriz de trazabilidad

Acontecimiento	Feature	Requisito
AC01	1	1, 4
AC02	1	5, 7
AC03	1	8
AC04	1	9
AC05	1	9
AC06	1	1
AC07	1	1
AC08	2	10
AC09	2	16
AC10	2	12, 13, 14
AC11	4	23
AC12	4	25
AC13	4	25
AC14	3	17, 22
AC15	3	17, 19, 20, 21
AC16	1	3, 5, 6, 7
AC17	1	3, 6, 7
AC18	1	3, 8
AC19	1	3, 8
AC20	1	3, 6
AC21	1	8
AC22	2, 3, 4	10, 21, 23

Tabla 2: matriz de trazabilidad entre features, requisitos y acontecimientos

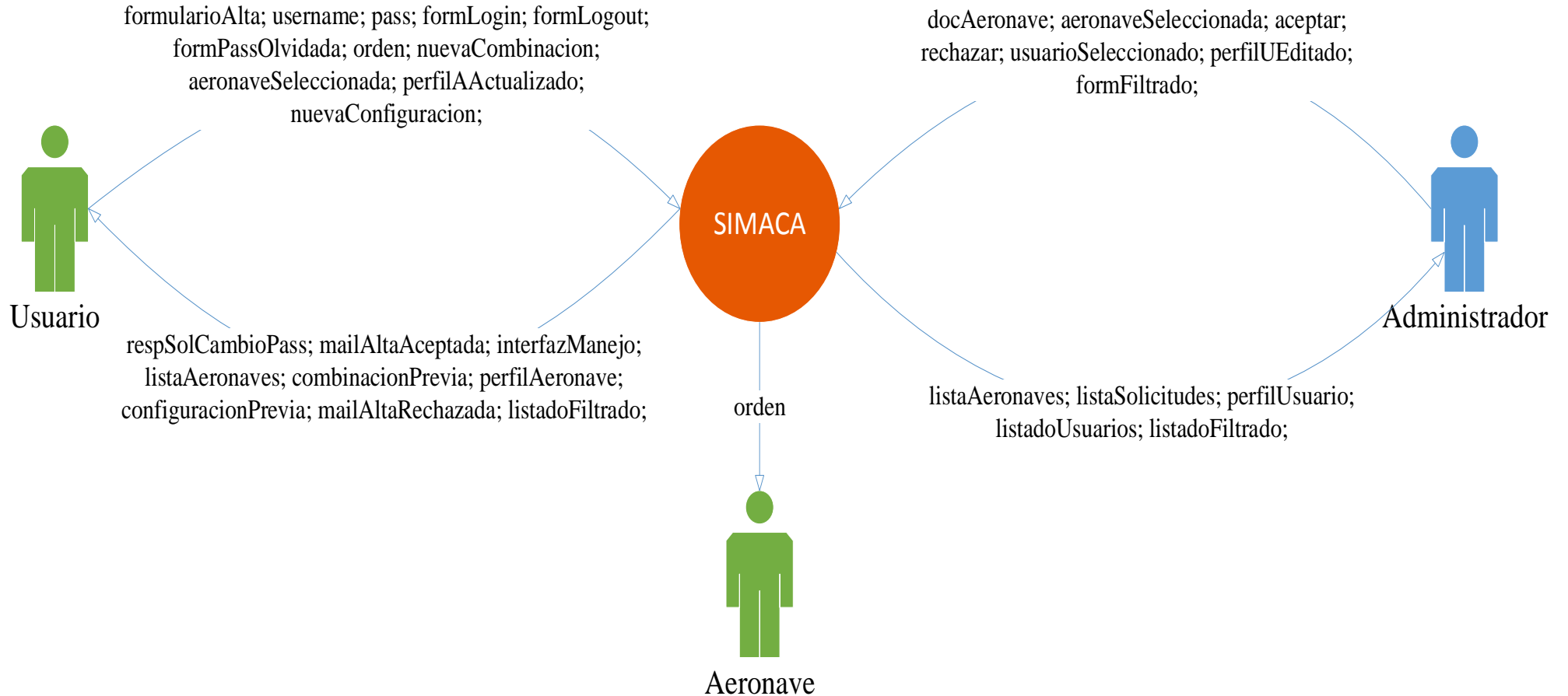
Acontecimiento	Burbuja
AC01	1.1.1
AC02	1.1.2
AC03	1.2
AC04	1.1.3
AC05	1.1.4
AC06	1.3
AC07	1.4
AC08	2.1
AC09	2.2
AC10	2.3
AC11	4.1
AC12	4.2
AC13	4.3
AC14	3.1
AC15	3.2
AC16	1.1.5
AC17	1.1.6
AC18	1.1.7
AC19	1.1.8
AC20	1.1.9
AC21	1.1.10
AC22	5

Tabla 3: Matriz de trazabilidad entre acontecimientos y burbujas

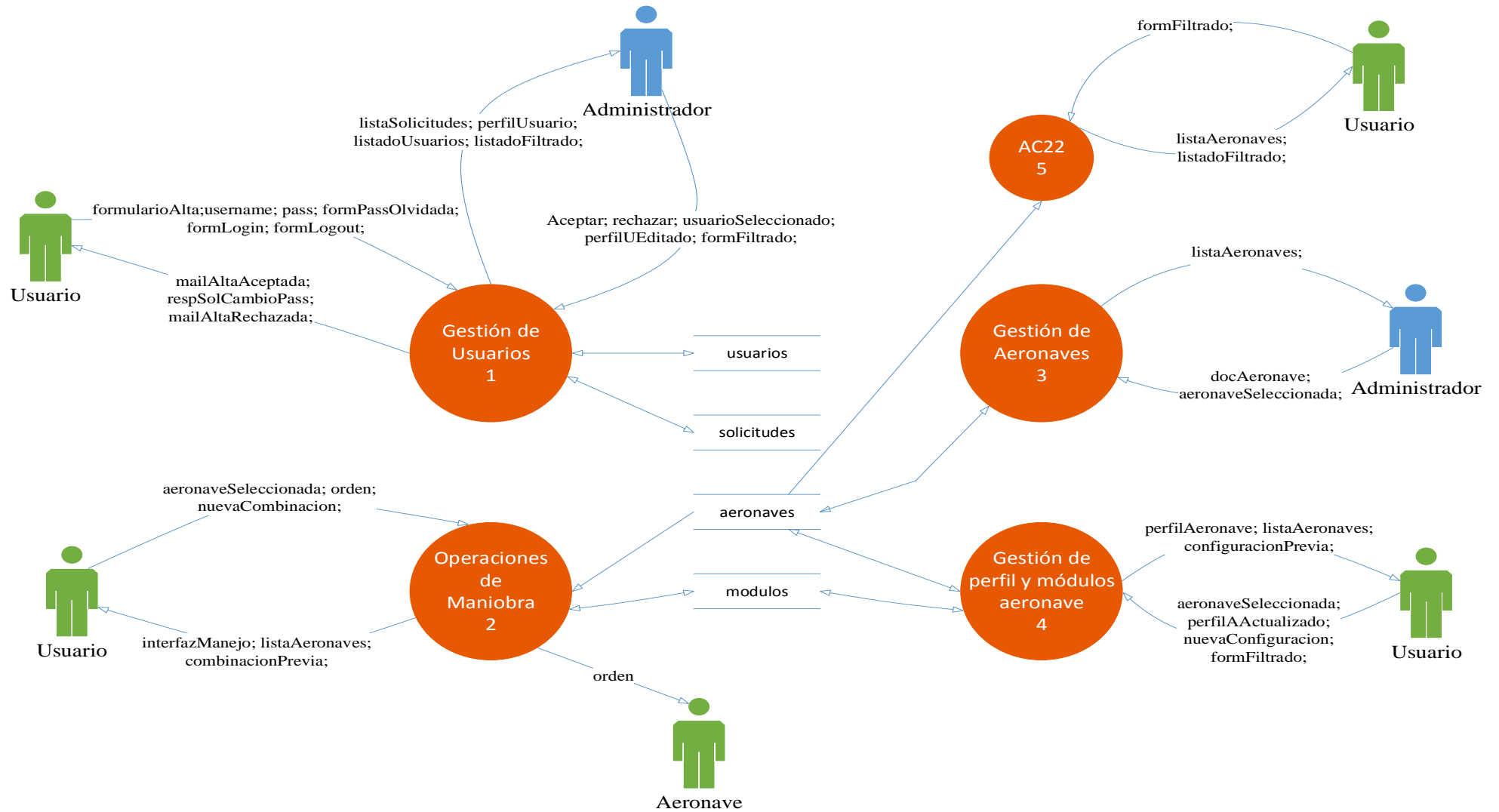
Modelo de Comportamiento

DFD

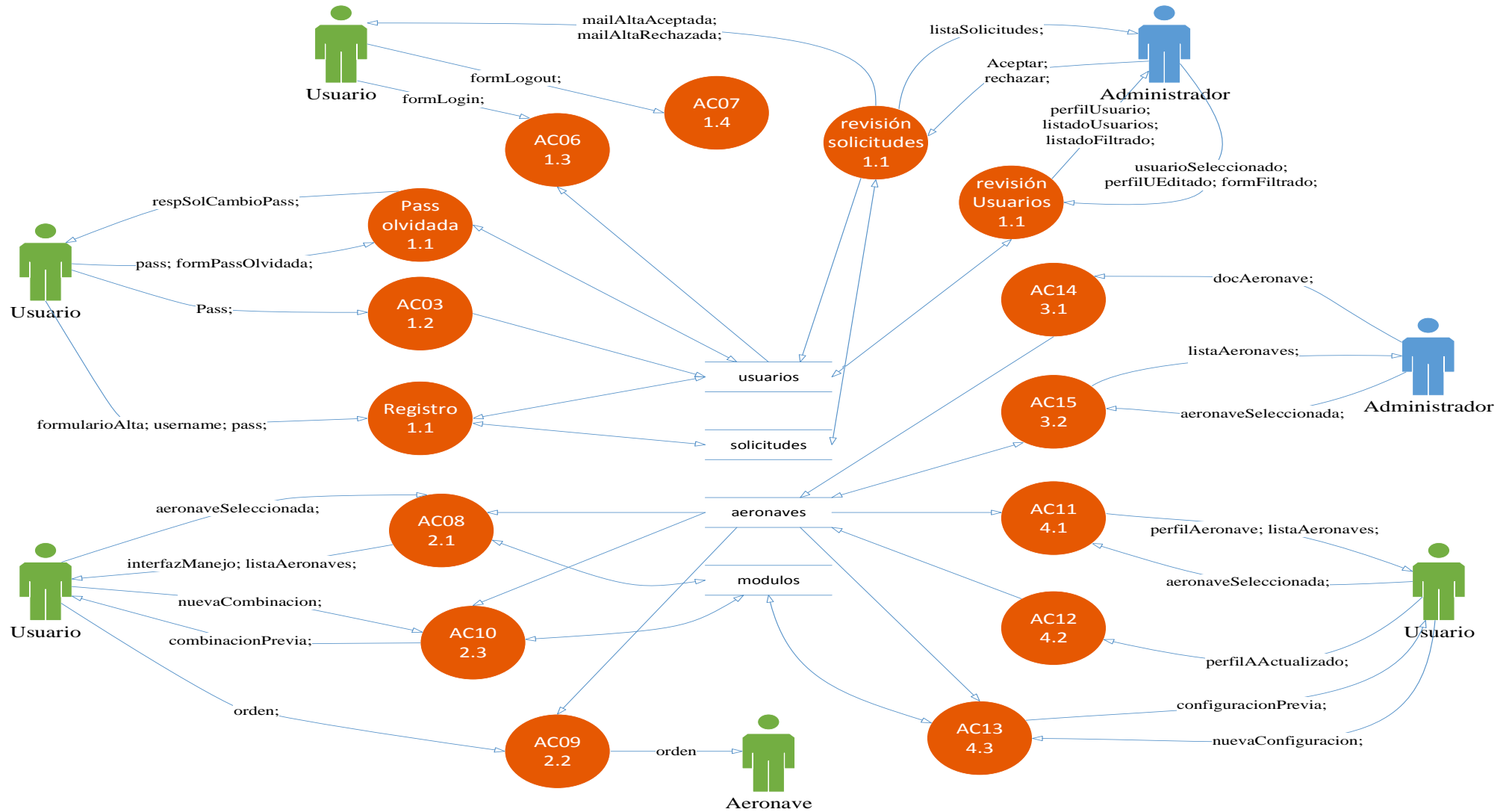
Nivel 0



Nivel 1



Nivel 2



Nivel 3

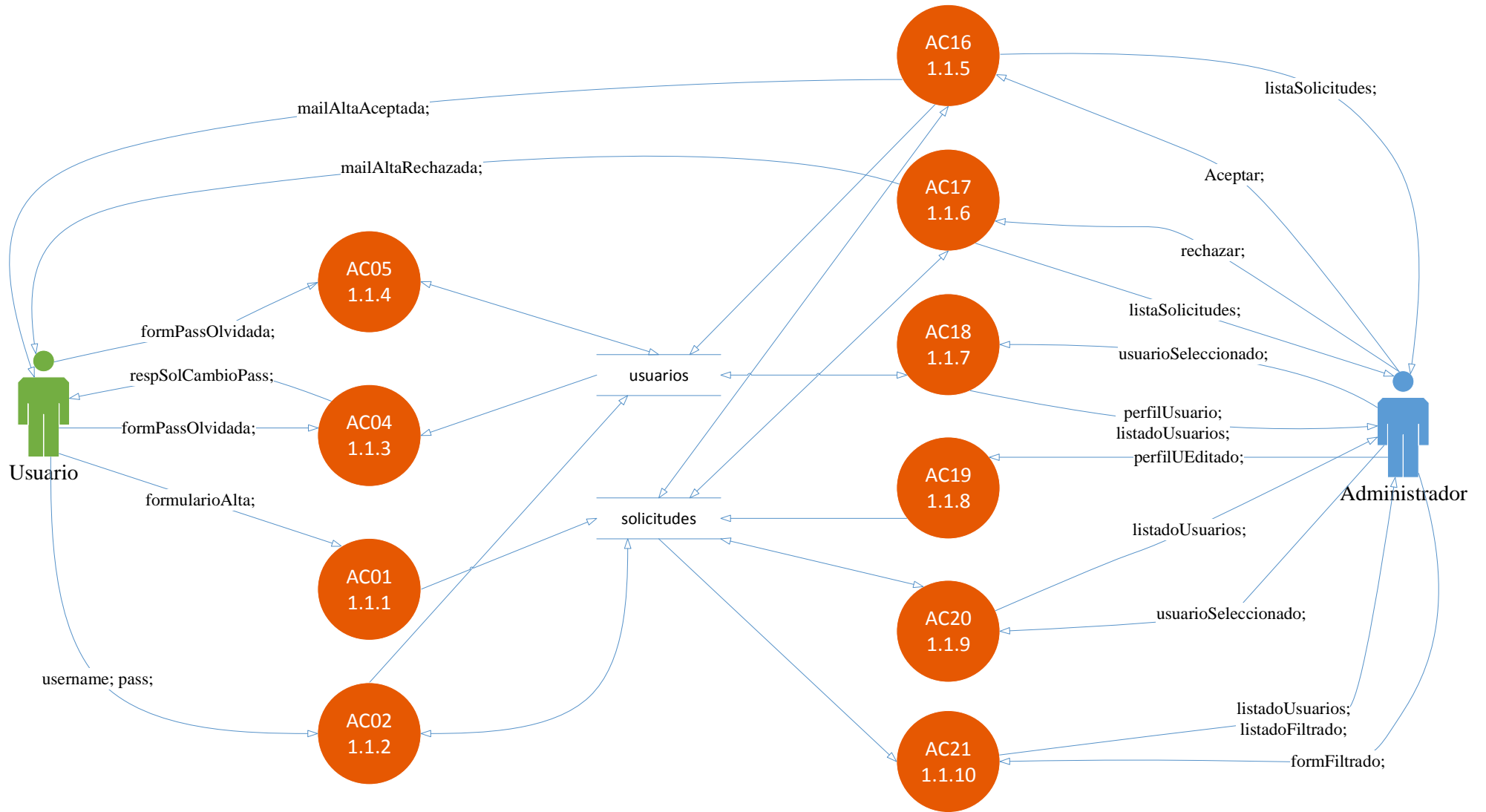
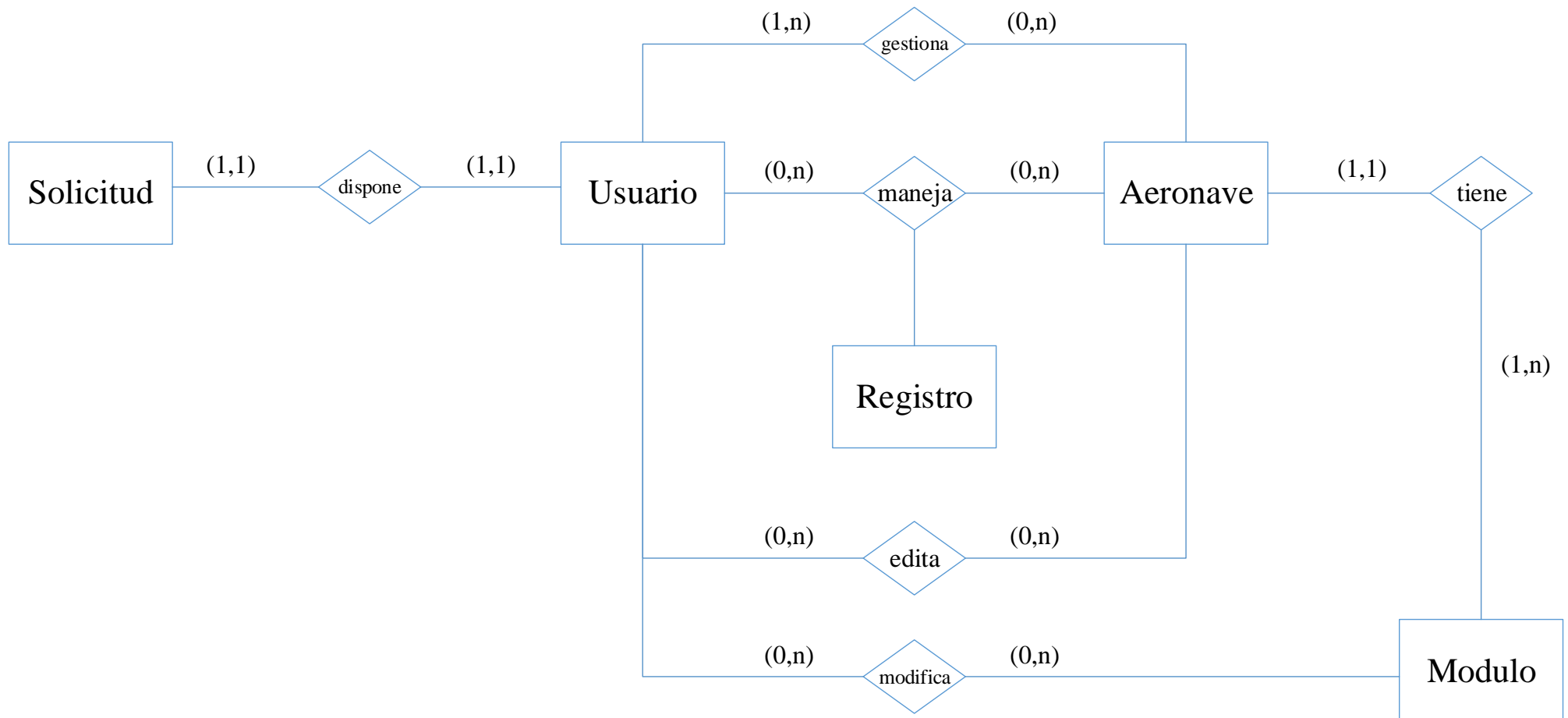


Diagrama de entidad-relación



Entidades:

- **Usuario:** Representa a la información que se recibe de la persona y los datos de la cuenta que va a utilizar dicha persona para acceder y utilizar los servicios del sistema.
- **Solicitud:** Representa a la información que se recibe de la persona interesada en acceder al sistema.
- **Aeronave:** Representa a la información de la aeronave física que va a ser controlada a través del sistema.
- **Modulo:** Representa a la información de los módulos instalados en la aeronave física relacionada.
- **Registro:** Es una entidad asociativa de la relación Usuario-Aeronave. En la cual se recoge información acerca de quién ha manejado qué aeronave en qué hora y fecha.

Relaciones:

- **Usuario-Aeronave:** Un usuario puede gestionar de 0 a n aeronaves; Una aeronave puede ser gestionada por 1 o n usuarios; Un usuario puede manejar de 0 a n aeronaves; Una aeronave puede ser manejada por 0 o n usuarios; Un usuario puede editar el perfil de 0 a n aeronaves; Una aeronave puede ser editada por 0 o n usuarios.
- **Aeronave-Modulo:** Una aeronave tiene 1 o n módulos instalados; Un módulo solo puede pertenecer a una aeronave.
- **Usuario-Modulo:** Un usuario puede modificar de 0 a n módulos; Un módulo puede ser editada por 0 o n usuarios.
- **Usuario-Solicitud:** Un usuario tiene que disponer de una solicitud y una solicitud sólo puede disponer de un único usuario.

DD

Solicitud = @idSolicitud + nombre + apellidos + dni + email + telefono + foto + estadoSolicitud + motivo

Usuario = @idUsuario + nombre + apellidos + dni + email + username + telefono + foto + pass + estadoUsuario

Aeronave = @idAeronave + nombre + pesoNeto + cargaUtil + nMotores + consumoMaximo + consumoMinimo + capacidadBateria + nCeldas + voltajeBateria + nBaterias + tipoConectorBateria + nModulos + nAlas + nTrenAterrizaje + estadoAeronave + pasoMotor + pasoGrados

Modulo = @idModulo + nombre + ip + puerto + nParametros + (Parametros) + nOperaciones + nombreOperaciones + nVersion + consumoPromedio + tipoConexion

Registro = @idRegistro + @idUsuario + @idAeronave + fecha + hora

nombre = *Es el nombre real del usuario, el que aparece en el DNI*

apellidos = *Los apellidos reales del usuario, los que aparecen en el DNI, siendo obligatorio el primero y el segundo opcional*

dni = *Es el número de DNI/NIE real del usuario con la/s letra/s incluida/s*

email = *Es una dirección de correo electrónico que utilice el usuario con normalidad. Es la dirección a la que se enviará toda comunicación por parte de la administración del sistema hacia el usuario*

motivo = *Es una explicación breve de la decisión tomada por el administrador, en caso de aceptar o denegar el registro del usuario*

username = *Es un sobrenombre que debe ser escogido por el usuario, de manera que sea único en el sistema y se usará para identificar al usuario en el proceso de login*

foto = *Es una dirección del tipo URI, en la que se encuentra una imagen descriptiva relacionada al usuario, la imagen estará alojada en el propio sistema*

estadoSolicitud = [nueva|pendiente|rechazada|aceptada]

estadoUsuario = [activo|administrador|cancelado]

estadoAeronave = [disponible|inhabilitado]

nombreOperaciones = {nombreDeOperacion}

nombreDeOperacion = *Es el nombre descriptivo y representativo de la operación a la que se quiere hacer referencia*

tipoConectorBateria = *Es el nombre del tipo de conector que utiliza la batería*

Parametros = *Se refiere a los posibles parámetros que se pueda pasar por el URL a la dirección resultante de la combinación de la IP y Puerto*

nVersion = *Es un número que indica la versión del software de la controladora que está instalada en la unidad controladora del módulo*

tipoConexion = *indica que la conexión establecida entre el módulo y el sistema se trata de información que sólo entra al sistema (e), sólo sale del sistema (s) o ambas opciones (a)*

fecha = *La estructura de la fecha ha de ser año-mes-día*

hora = *Se utilizará el formato hora:minutos:segundos*

pasoMotor = *Se define como la unidad mínima de incremento en unidades de porcentajes para el aumento de potencia de los motores*

pasoGrados = *Se define como la cantidad mínima de giro realizable por acción de giro respecto a un eje, medido en grados*

Unidad controladora del cuadricóptero

La unidad controladora consistirá en el tratamiento de la recepción de las órdenes de operaciones de maniobras, tales como avanzar, retroceder, girar, etc. También consistirá en la estabilización de la propia aeronave. Por tanto, esta unidad se encarga de establecer, mantener y actualizar los parámetros de vuelo, que son los grados de inclinación y la potencia de giro de los motores.

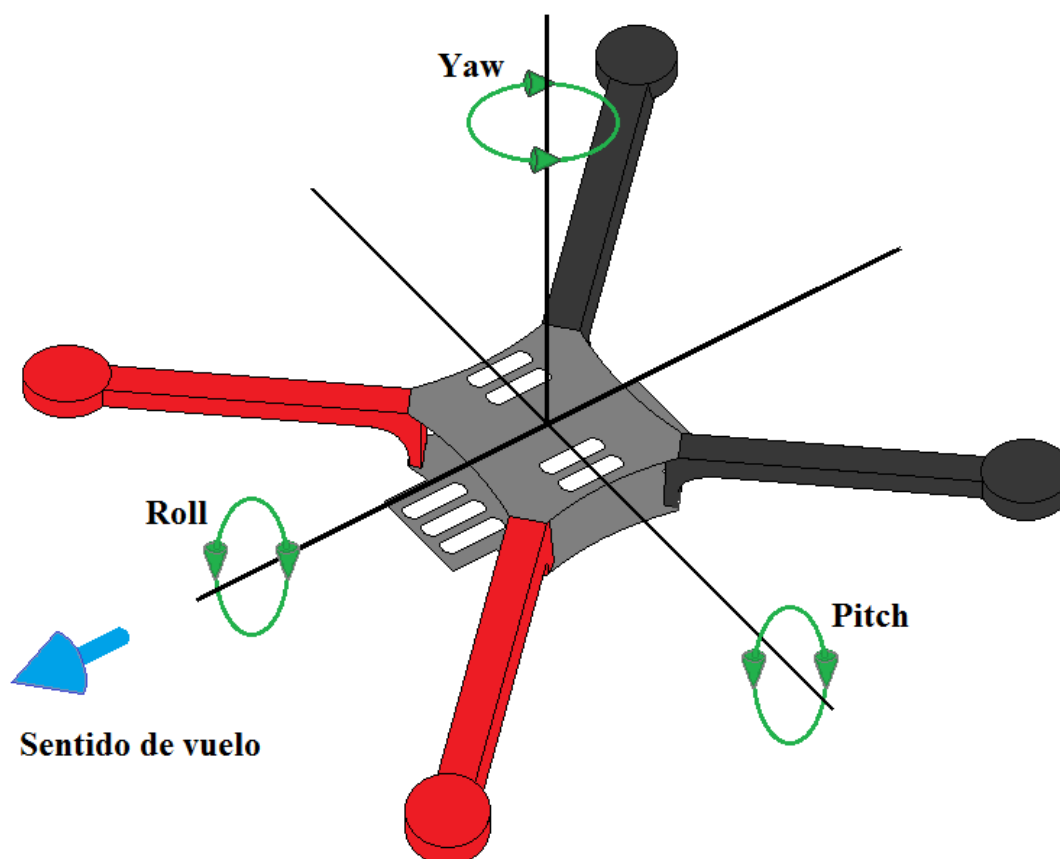


Figura 37: Yaw, Pitch y Roll

En este apartado cabe definir los términos de *Pitch*, *Roll* y *Yaw*. Los tres términos expresan grados de giro respecto de los 3 posibles ejes, como se puede observar en la [Figura 37](#). *Pitch* mide los grados de giro en el eje longitudinal respecto al plano horizontal, logrando identificar si se está avanzando o retrocediendo. *Roll* mide los grados de giro en el eje transversal respecto del plano horizontal, permitiendo la identificación de si se está desplazando hacia izquierda o derecha. *Yaw* mide los grados de giro del plano de la aeronave respecto a su eje vertical, logrando así identificar si se está realizando un giro hacia izquierda o derecha.

Existen parámetros de vuelo que no serán controlables directamente, sino que son informativos y quedan a disposición del piloto el cómo interpretarlos, como son la altura, la capacidad de la batería restante estimada, la velocidad de vuelo o la temperatura ambiental.

La unidad controladora manejará los incrementos de la potencia de los motores por pasos (será conocido como `pasoMotor`), se conoce como paso la diferencia de la unidad mínima de incremento o decremento de la potencia de cada motor. En este caso, el paso tiene un valor comprendido entre el 0 y un valor, a establecer, como límite de resolución de los incremento. Este límite de resolución tiene objetivo de escalar el paso de un intervalo, establecido entre 0 y el límite de resolución, a un intervalo dado de las frecuencias en las que trabaja los controladores electrónicos de velocidad (En inglés, ESC). No obstante, dichos valores pueden ser cambiados desde el perfil de la aeronave.

Por ejemplo:

Suposiciones iniciales:

`pasoMotor=1;`

`intervalo del pasoMotor=[0,2300];`

`intervalo de las frecuencias de funcionamiento de los ESC=[750, 1900];`

Resolución del paso:

Distancia real: $1900-750 = 1150$;

Distancia virtual: $2300-0 = 2300$;

Resolución resultante: **$1150/2300 = 0,5$**

Resultado:

Por cada unidad de incremento con valor `pasoMotor` en el intervalo [0, 2300] supone un incremento de 0,5 en el intervalo [750, 1900], logrando una sensación de suavidad aumentada por la mitad.

La implementación de la unidad controladora funcionará dentro de un bucle, pero en primer lugar, se debe de inicializar algunos servicios en el ámbito de *setup*, que corresponde a:

1. Sincronizar los ESC.
2. Configurar los sensores que deben recoger los parámetros de Pitch, Roll y Yaw (son recogidos por el módulo MPU).

3. Conectar con el router WiFi.
4. Arrancar el servidor de sockets.

Una vez realizado el ámbito del *setup*, se inicia un bucle infinito en el que se ejecutará las acciones principales de la aeronave, que será los siguientes pasos definidos:

1. Reconocer que hay un cliente en el servidor creado.
2. Si no hay cliente conectado, la aeronave debe mantener el último estado de vuelo.
3. Si hay cliente conectado se debe esperar a recibir algo por el socket.
4. Si no se recibe nada del socket, entonces se debe mantener el último estado de vuelo.
5. Si se ha recibido algo, entonces se debe reconocer el tipo de operación y proceder a actuar en consecuencia.
6. Si no se reconoce el tipo de operación, se debe mantener el último estado de vuelo.

Protocolo de comunicación

Las comunicaciones entre la aeronave y el sistema, deberán cumplir las siguientes especificaciones:

Para las comunicaciones de operaciones de maniobra. Cada pulsación de tecla o pulsación de botón de la interfaz web, se envía un mensaje con el siguiente formato estándar: “**op:**[nombre operación]\”, siendo [nombre operación] las siguientes posibles opciones: dsp (despegar); ate (aterrizar); ava (avanzar); ret (retroceder); izq (izquierda); der (derecha); giz (girar hacia izquierda), gde (girar hacia derecha), ele (elevar), dsc (descender) y stp (para de emergencia).

En las comunicaciones de especificación de los parámetros de pasoMotor y pasoGrados, han de cumplir los siguientes formatos: En el caso del pasoMotor es “**pm:**[valor]\”, donde [valor] es el valor correspondiente a la unidad mínima de incremento de la potencia de cada motor en unidades de porcentaje.

Las comunicaciones de información de la aeronave al sistema ha de tener el siguiente formato estándar: “**inf:**[nombre de parámetro]=[valor]\”, donde [nombre de parámetro] corresponde al nombre del parámetro que se quiere informar al sistema, pudiendo ser: alt (altura) o bat (batería). Donde [valor] corresponde al valor del parámetro que ha de tener como máximo 5 caracteres numéricos, puede llevar como máximo un carácter de puntuación “,” o “.” para distinguir decimales. Estas informaciones serán generadas automáticamente según el tiempo establecido por cada módulo correspondiente.

ANEXO D: DISEÑO DE BAJO NIVEL

Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino

Diseño de Bajo Nivel

Autor: GenLei Cui Ji

Tutor: Víctor Nieto Lluís

Fecha: 12/03/2014

Índice

Introducción	D-2
Consideraciones	D-2
Tablas	D-3
Especificación de procesos	D-5
Unidad controladora del cuadricóptero	D-11

Introducción

El presente documento tiene la finalidad de describir el diseño de bajo nivel, en adelante se conocerá como DBN, que consiste en especificar con mayor detalle los atributos de cada entidad del modelo Entidad-Relación de la base de datos, propuesto en el documento de diseño de alto nivel, en adelante DAN, así como describir, implementar y especificar con pseudocódigos a cada burbuja o acontecimiento descrito en el documento DAN.

Adicionalmente, se definirá el diseño de la unidad controladora de la aeronave prototipo utilizada para realizar las pruebas con el sistema.

Consideraciones

En esta sección se detallarán las razones de algunas decisiones tomadas para el desarrollo del DBN.

Los tamaños de los tipos se han especificado teniendo en cuenta las futuras ampliaciones y estableciendo la unidad métrica sobre la que utilizará el sistema para interpretar dicha información. Por ejemplo, en el caso de los consumos y capacidades de las baterías se medirán en mAh. Se ha optado por establecer 7 cifras de enteros positivos porque el número motores y/o módulos pueden variar mucho de una aeronave a otra y se ha establecido que como máximo el consumo conjunto podrá rondar alrededor de 7 cifras.

Algunos datos del tipo texto como “parámetros” y “nombreOperaciones” se pretende que sean una lista de elementos. Para identificar cada elemento de los datos anteriores, se utilizará un separador para que el sistema los reconozca unitariamente. El separador será el carácter “;”.

Tablas

Nombre	Tipo	Nulo	Predeterminado	Extra
idSolicitud	Unsigned int(5)	No		Primary Key; Auto_Increment;
nombre	varchar(35)	No		
apellidos	varchar(30)	No		
dni	varchar(10)	No		UNIQUE;
email	varchar(50)	No		UNIQUE;
teléfono	Unsigned int(9)	No		
foto	text	No		
estadoSolicitud	ENUM('nueva', 'pendiente', 'aceptada', 'rechazada')	No	nueva	
motivo	text	Sí		

Tabla 4: Definición de la tabla de Solicitud

Nombre	Tipo	Nulo	Predeterminado	Extra
idUsuario	Unsigned int(5)	No		Primary Key; Auto_Increment;
nombre	varchar(35)	No		
apellidos	varchar(30)	No		
dni	varchar(10)	No		UNIQUE
email	varchar(50)	No		UNIQUE
username	varchar(30)	No		UNIQUE
telefono	Unsigned int(9)	No		
foto	text	No		
pass	varchar(30)	No		
estadoUsuario	ENUM('activo', 'administrador', 'cancelado')	No	activo	

Tabla 5: Definición de la tabla de Usuario

Nombre	Tipo	Nulo	Predeterminado	Extra
idModulo	Unsigned int(5)	No		Primary Key; Auto_Increment;
nombre	varchar(30)	No		UNIQUE
ip	varchar(15)	No		UNIQUE
puerto	Unsigned int(2)	No		
nParametros	Unsigned int(3)	No		
parametros	text	No		
nOperaciones	Unsigned int(3)	No		
nombreOperaciones	text	No		
nVersion	varchar(5)	No		
consumoPromedio	Unsigned int(7)	No		
tipoConexion	ENUM('e', 's', 'a')	No	a	

Tabla 6: Definición de la tabla de Modulo

Nombre	Tipo	Nulo	Predeterminado	Extra
idRegistro	Unsigned int(5)	No		Primary Key; Auto_Increment;
idUsuario	Unsigned int(5)	No		FOREIGN KEY
idAeronave	Unsigned int(5)	No		FOREIGN KEY
fecha	DATE	No		
hora	TIME	No		

Tabla 7: Definición de la tabla de Registro

Nombre	Tipo	Nul o	Predeterminad o	Extra
idAeronave	Unsigned int(5)	No		Primary Key; Auto_Increment ;
nombre	varchar(30)	No		UNIQUE
pesoNeto	Unsigned int(7)	No		
cargaUtil	Unsigned int(7)	No		
nMotores	Unsigned int(2)	No		
consumoMaximo	Unsigned int(7)	No		
consumoMinimo	Unsigned int(7)	No		
capacidadBateria	Unsigned int(7)	No		
nCeldas	Unsigned int(2)	No		
voltajeBateria	float(5,2)	No		
nBaterias	Unsigned int(2)	No		
tipoConectorBateria	varchar(20)	No		
nModulos	Unsigned int(2)	No		
nAlas	Unsigned int(2)	No		
nTrenAterrizaje	Unsigned int(2)	No		
estado	ENUM('disponible', 'inhabilitado')	No		
pasoMotor	Unsigned int(2)	No	1	
pasoGrados	float(2,1)	No	1	

Tabla 8: Definición de la tabla de Aeronave

Especificación de procesos

Proceso	1.1.1 (AC01)
Descripción	El usuario envía el formulario de alta en el sistema.
Entrada	formularioAlta
Operación	1- Volcar los datos recibidos en la tabla de solicitud; 1.1- En el campo de la foto poner la dirección de la foto, utilizando de nombre el identificador de la solicitud. 1.2- Se le asigna el valor “nueva” al estado de solicitud. 2- Guardar la foto del usuario con el idSolicitud como nombre, en el directorio de las fotos.
Salida	-

Tabla 9: Especificación del proceso 1.1.1

Proceso	1.1.2 (AC02)
Descripción	El usuario crea una contraseña y nombre de usuario para su nueva cuenta.
Entrada	username; pass;
Operación	1- Comprobar que el enlace es legítimo y tiene asociado a una solicitud aceptada. 2- Crear una nueva entrada en la tabla de usuarios con los datos de la solicitud asociada y utilizando username y pass como datos de acceso a la cuenta.
Salida	-

Tabla 10: Especificación del proceso 1.1.2

Proceso	1.2 (AC03)
Descripción	El usuario cambia la contraseña de su cuenta desde su perfil.
Entrada	pass;
Operación	1- Comprobar que los dos campos de la nueva contraseña coinciden. 1.1- En caso de coincidir, efectuar el cambio en la base de datos 1.2- En otros casos mostrar un mensaje con el error correspondiente.
Salida	-

Tabla 11: Especificación del proceso 1.2

Proceso	1.1.3 (AC04)
Descripción	El usuario solicita reasignar la contraseña de su cuenta por olvido.
Entrada	formPassOlvidada;
Operación	1- Comprobar si el correo electrónico recibido en el formulario formPassOlvidada corresponde a algún usuario registrado. 1.1- Si se encuentra coincidencia, se procede a enviar un correo a la dirección de correo electrónico del usuario en cuestión. 1.2- En otro caso, mostrar un mensaje de error al usuario.
Salida	respSolCambioPass;

Tabla 12: Especificación del proceso 1.1.3

Proceso	1.1.4 (AC05)
Descripción	El usuario asigna la nueva contraseña de su cuenta por olvido.
Entrada	pass;
Operación	1- Comprobar que los campos de la nueva contraseña coinciden. 1.1- Si coinciden se procede a cambiarlo en la base de datos. 1.2- En otros casos devolver un error.
Salida	-

Tabla 13: Especificación del proceso 1.1.4

Proceso	1.3 (AC06)
Descripción	El usuario se autentica en el sistema.
Entrada	formLogin;
Operación	1- Comprobar los datos de acceso con la base de datos 1.1- Si se encuentra con una y sola una coincidencia, entonces permitir iniciar la sesión. 1.2- En otros casos mostrar mensaje de error.
Salida	-

Tabla 14: Especificación del proceso 1.3

Proceso	1.4 (AC07)
Descripción	El usuario cierra su sesión en el sistema.
Entrada	formLogout;
Operación	1- Se borran todos los datos guardados en la sesión del usuario. 2- Se cierra la sesión del usuario.
Salida	-

Tabla 15: Especificación del proceso 1.4

Proceso	1.1.5 (AC16)
Descripción	El administrador revisa y acepta la solicitud de un alta.
Entrada	aceptar;
Operación	1- Si se recibe aceptar, entonces enviar un correo de confirmación de registro al usuario, actualizar el motivo de la resolución y cambiar el estado de la solicitud a aceptada. 2- Si se recibe un pendiente, entonces cambiar el estado de la solicitud a pendiente.
Salida	listaSolicitudes; mailAltaAceptada;

Tabla 16: Especificación del proceso 1.1.5

Proceso	1.1.6 (AC17)
Descripción	El administrador revisa y rechaza la solicitud de un alta.
Entrada	rechazar;
Operación	1- Si se recibe rechazar, entonces enviar un correo de rechazo de registro al usuario, actualizar el motivo de la resolución y cambiar el estado de la solicitud a rechazada. 2- Si se recibe un pendiente, entonces cambiar el estado de la solicitud a pendiente.
Salida	listaSolicitudes; mailAltaRechazada;

Tabla 17: Especificación del proceso 1.1.6

Proceso	1.1.7 (AC18)
Descripción	El administrador selecciona el perfil de un usuario a modificar.
Entrada	usuarioSeleccionado;
Operación	1- Mostrar un listado de todos los usuarios existentes en el sistema 3- Buscar el perfil del usuario seleccionado. 2- Mostrar todos los datos del perfil del usuario seleccionado
Salida	perfilUsuario; listadoUsuarios;

Tabla 18: Especificación del proceso 1.1.7

Proceso	1.1.8 (AC19)
Descripción	El administrador edita el perfil de un usuario.
Entrada	perfilUEditado;
Operación	1- Actualizar los datos recibidos del perfil del usuario
Salida	-

Tabla 19: Especificación del proceso 1.1.8

Proceso	1.1.9 (AC20)
Descripción	El administrador elimina un usuario del sistema.
Entrada	usuarioSeleccionado;
Operación	1- Mostrar un listado de todos los usuarios existentes en el sistema. 2- Establecer el estado de la cuenta del usuario seleccionado a cancelado.
Salida	listadoUsuarios;

Tabla 20: Especificación del proceso 1.1.9

Proceso	1.1.10 (AC21)
Descripción	El administrador filtra los resultados de un listado de usuarios.
Entrada	formFiltrado;
Operación	1- Se recibe un formulario de filtro, con el que se recoge un nombre de usuario. 2- Se filtran los resultados de la lista de todos los usuarios del sistema, que cumplan con el parámetro de filtro.
Salida	listadoFiltrado; listadoUsuarios;

Tabla 21: Especificación del proceso 1.1.10

Proceso	2.1 (AC08)
Descripción	El usuario selecciona una aeronave para manejarla.
Entrada	aeronaveSeleccionada;
Operación	1- Mostrar un listado de todas las aeronaves en estado disponible en el sistema. 2- Se busca el perfil de la aeronave seleccionada. 3- Mostrar la interfaz de manejo de la aeronave seleccionada.
Salida	listaAeronaves; interfazManejo;

Tabla 22: Especificación del proceso 2.1

Proceso	2.2 (AC09)
Descripción	El usuario envía órdenes de maniobras a la aeronave.
Entrada	orden;
Operación	1- Recibir la orden y enviarla al conjunto de IP y puerto correspondiente al módulo de operaciones de maniobras de la aeronave en cuestión.
Salida	orden;

Tabla 23: Especificación del proceso 2.2

Proceso	2.3 (AC10)
Descripción	El usuario personaliza la combinación de teclas para el manejo de la aeronave desde la interfaz de manejo.
Entrada	nuevaCombinacion;
Operación	1- Se muestra la configuración previa de la combinación de teclas. 2- Actualizar la nueva configuración de combinación de teclas.
Salida	combinacionPrevia;

Tabla 24: Especificación del proceso 2.3

Proceso	3.1 (AC14)
Descripción	El administrador añade una aeronave al sistema.
Entrada	docAeronave;
Operación	1- Se sube el fichero al sistema. 2- Se lee el fichero y se reconocen los datos en ella. 3- Se crea una nueva entrada de perfil de aeronave y se guardan los datos relativos a la aeronave. 4- Se lee la variable de número de módulos instalados en la aeronave. Se inicia un bucle para crear la misma cantidad de nuevas entradas de módulos relacionados con la aeronave en cuestión. 5- Se guardan los datos leídos de cada módulo en la entrada correspondiente. 6- Se cierra el fichero.
Salida	-

Tabla 25: Especificación del proceso 3.1

Proceso	3.2 (AC15)
Descripción	El administrador elimina una aeronave del sistema.
Entrada	aeronaveSeleccionada;
Operación	1- Se muestra una lista de todas las aeronaves en el sistema. 2- Se actualiza el estado de la aeronave a inhabilitado y ésta se queda inaccesible para los usuarios.
Salida	listaAeronaves;

Tabla 26: Especificación del proceso 3.2

Proceso	4.1 (AC11)
Descripción	El usuario selecciona el perfil de una aeronave.
Entrada	aeronaveSeleccionada;
Operación	1- Mostrar un listado de todas las aeronaves en estado disponible en el sistema. 2- Se muestra todos los datos del perfil de la aeronave seleccionada.
Salida	listaAeronaves; perfilAeronave;

Tabla 27: Especificación del proceso 4.1

Proceso	4.2 (AC12)
Descripción	El usuario edita el perfil de una aeronave.
Entrada	perfilAActualizado;
Operación	1- Se actualizan los datos recibidos del perfil de la aeronave correspondiente.
Salida	-

Tabla 28: Especificación del proceso 4.2

Proceso	4.3 (AC13)
Descripción	El usuario modifica las configuraciones de un módulo determinado desde el perfil de la aeronave.
Entrada	nuevaConfiguracion;
Operación	1- Se muestra la configuración previa del módulo correspondiente. 2- Se actualizan los datos de configuración del módulo.
Salida	configuracionPrevia;

Tabla 29: Especificación del proceso 4.3

Proceso	5 (AC22)
Descripción	El usuario filtra los resultados de un listado de aeronaves.
Entrada	formFiltrado;
Operación	1-Se muestra un formulario con parámetro de filtro el nombre de aeronave. 2- Se muestra los resultados a partir de filtrar la lista de aeronaves con el parámetro de filtro.
Salida	listaAeronaves; listadoFiltrado;

Tabla 30: Especificación del proceso 5

Unidad controladora del cuadricóptero

Inicialización:

```

For all ESC {
    Max_señal;
}
For all ESC {
    Min_señal;
}
Iniciar_MPU();
Conectar_router_wifi();

```


Bucle infinito:

```
Si_hay_cliente() {
    Actulizar_pitch_roll_yaw();
Si_recibe_mensaje() {
    If (mensaje == operacion1) {
        Código_operacion = 1;
    } else if (mensaje == operacion2) {
        Código_operacion = 2;
    } ...

    Switch (código_operacion) {
        Case operacion:    porcentajePotencia    =
{a,b,c,d}
                                Break;
        ...
        Default:         //no hacer nada
    }
    Ajustar_portencia_motores(porcentajePotencia);
}
Si_no_recibe_mensaje() {
    Mantener_ultimo_estado();
}
}
No_hay_cliente() {
    Mantener_ultimo_estado();
}
```

ANEXO E: MANUAL DE INSTRUCCIONES

Diseño de un sistema para el manejo a distancia de aeronaves construidas con arduino

Manual de Instrucciones

Autor: GenLei Cui Ji

Tutor: Víctor Nieto Lluís

Fecha: 22/05/2014

Índice

Introducción	E-3
Consideraciones	E-3
Instalación	E-4
Configurar entorno de ejecución	E-4
Instalación del servidor	E-5
Seguridad	E-11
Base de datos	E-12
Uso	E-13
Conexión	E-13
Desconexión	E-14

Introducción

Este manual de instrucciones tiene como principal objetivo describir los pasos necesarios para la correcta instalación del sistema implementado. Adicionalmente, se darán algunas directrices sobre el mantenimiento del sistema, así como su uso y algunas particularidades.

Consideraciones

SIMACA está implementado sobre la plataforma de servidor XAMPP, concretamente la versión 1.8.3-3 Portable, que viene conjuntamente con algunas herramientas y tecnologías, tales como Apache versión 2.4.9; MySQL versión 5.6.16; PHP versión 5.5.11; phpMyAdmin versión 4.1.12; OpenSSL versión 1.0.1 y XAMPP Control Panel versión 3.2.1.

Para la gestión automática de envíos de correos electrónicos desde la página web implementado en el sistema se ha utilizado el proyecto PHPMailer versión 5.2.7.

El lenguaje de programación utilizado para el pre proceso, es decir, el código que se ejecuta en la parte del servidor, de las páginas web ha sido PHP, se ha empleado métodos y funciones del propio lenguaje que están soportados hasta la versión 5.5 de PHP y 5.6 de MySQL. Cabe mencionar aquellas funciones y librerías propias del lenguaje PHP críticas para el correcto funcionamiento del sistema y que hay que vigilar por si se decide actualizar o cambiar de versión del servidor XAMPP, que son:

- **mysqli:** La librería mysqli de PHP es esencial para el correcto funcionamiento del acceso con la base de datos, así como todas las peticiones a posteriori.
- **crypt():** Esta función es esencial para generar y almacenar información cifrada, de tal forma que ni siquiera los administradores puedan reconocer el contenido.
- **md5():** Esta función es utilizada para generar códigos y nombres de fotos con un formato corto, que previamente han sido cifrados con la función crypt().

Los lenguajes de maquetación escogidos para desarrollar las páginas web han sido HTML5 y CSS3. La implementación de cada página se ha dividido en dos partes, la parte del documento HTML y en otro documento el código de CSS, que es incorporado en el de HTML con una referencia.

Por último, se ha utilizado JavaScript y la librería jQuery 2.1.1, dado por el carácter de ejecución local del JavaScript, permite manejar los elementos del documento DOM (en inglés, *Document Object Model*), que es el documento generado y manejado por el navegador para mostrar la interfaz web. Además de permitir el manejo de DOM, puede capturar las pulsaciones de teclas y con la ayuda de la librería jQuery, es capaz de establecer comunicaciones *WebSocket*.

Instalación

Configurar entorno de ejecución

El proceso de instalación para el desarrollo se efectúa en un entorno de red local intranet con un equipo funcionando bajo Windows 8.1, con una redirección de la URL www.simaca.com hacia el “localhost”. Para ello realizar los siguientes pasos:

1. Abrir un bloc de notas con derechos de ejecución de administrador.
2. Abrir el documento localizado en: “C:\Windows\System32\drivers\etc\hosts”.
3. El documento debe de tener un contenido similar al que aparece en la [Figura 38](#):

```
1 # Copyright (c) 1993-2009 Microsoft Corp.
2 #
3 # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4 #
5 # This file contains the mappings of IP addresses to host names. Each
6 # entry should be kept on an individual line. The IP address should
7 # be placed in the first column followed by the corresponding host name.
8 # The IP address and the host name should be separated by at least one
9 # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #      102.54.94.97      rhino.acme.com          # source server
17 #      38.25.63.10      x.acme.com              # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1          localhost
21 #   ::1                localhost
```

Figura 38: Contenido por defecto del documento hosts

4. Añadir las siguientes líneas en el documento, tal y como se muestra en la [Figura 39](#):
localhost www.simaca.com
localhost simaca.com
127.0.0.1 simaca.com
127.0.0.1 www.simaca.com

```

1  # Copyright (c) 1993-2009 Microsoft Corp.
2  #
3  # This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
4  #
5  # This file contains the mappings of IP addresses to host names. Each
6  # entry should be kept on an individual line. The IP address should
7  # be placed in the first column followed by the corresponding host name.
8  # The IP address and the host name should be separated by at least one
9  # space.
10 #
11 # Additionally, comments (such as these) may be inserted on individual
12 # lines or following the machine name denoted by a '#' symbol.
13 #
14 # For example:
15 #
16 #       102.54.94.97       rhino.acme.com           # source server
17 #       38.25.63.10       x.acme.com               # x client host
18
19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1       localhost
21 #   ::1             localhost
22 localhost       www.simaca.com
23 localhost       simaca.com
24 127.0.0.1       simaca.com
25 127.0.0.1       www.simaca.com

```

Figura 39: Nuevas líneas añadidas al documento hosts

5. Guardar y cerrar el block de notas.

Instalación del servidor

La instalación del servidor XAMPP se inicia con la descarga del [instalador](#) o bien su [versión portable](#) (la escogida para el desarrollo del proyecto). Cabe mencionar que durante el desarrollo del proyecto, se hizo patente un fallo de seguridad de la librería OpenSSL, incorporada en este servidor XAMPP, por lo que es conveniente instalar [el paquete de parche](#) para resolver dicho fallo. Para comenzar con la instalación del servidor basta con seguir los pasos que se indican en el instalador o en su defecto, para la versión portable, siga los siguientes pasos:

1. Descomprimir todos los ficheros en un directorio cualquiera. Se va a considerar que se ha descomprimido en el directorio “C:\xampp”, para esclarecer los pasos.
2. Ejecutar el fichero, haciendo doble clic sobre el documento “setup_xampp.bat”, localizado en “C:\xampp”. Debería aparecer el contenido mostrado en la [Figura 40](#).

```
#####
#
#               XAMPP USB Stick Installation
#
#####

#####
# ApacheFriends XAMPP setup win32 Version
#-----#
# Copyright (c) 2002-2014 ApacheFriends 1.8.3
#
#-----#
# Authors: Kay Vogelgesang <kvo@apachefriends.org>
#          Carsten Wiedmann <webmaster@wiedmann-online.de>
#####

Configure XAMPP with awk for 'Windows_NT'
Updating configuration files ... please wait ... DONE!

##### Have fun with ApacheFriends XAMPP! #####

Press any key to continue . . .
```

Figura 40: Mensaje de éxito en la configuración inicial del entorno de ejecución de XAMPP

3. Ejecutar el fichero “xampp-control.exe” localizado en el mismo directorio. Se abrirá una pantalla similar a la mostrada en la [Figura 41](#).

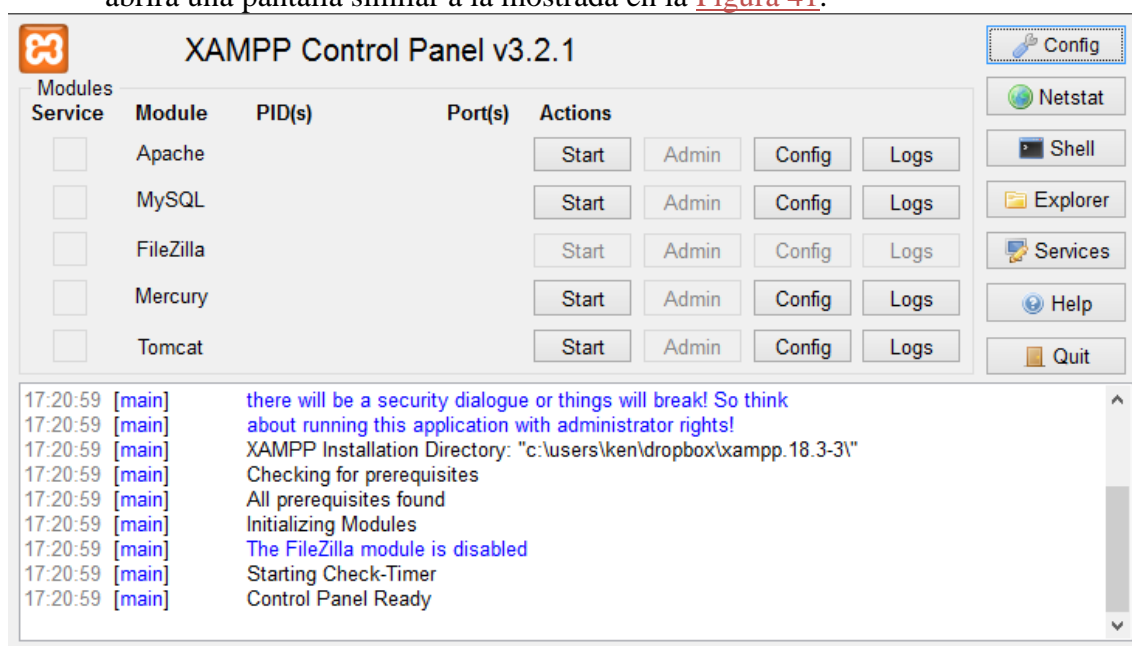



Figura 41: Panel de control de XAMPP

4. Inicie los servicios Apache y MySQL, haciendo clic sobre los botones “Start” correspondientes. En el panel de control debe reflejar el contenido mostrado en la [Figura 42](#). Si salen avisos de seguridad del Firewall de Windows, como los mostrados en la [Figura 43](#) y [Figura 44](#), se han de permitir, al menos, las comunicaciones en el ámbito de las redes locales privadas.



XAMPP Control Panel v3.2.1

Modules				
Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	6412 6828	80, 443	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/>
<input type="checkbox"/>	MySQL	7036	3306	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/>

17:20:59 [main] Initializing Control Panel
17:20:59 [main] Windows Version: Windows 8 Pro 64-bit
17:20:59 [main] XAMPP Version: 1.8.3
17:20:59 [main] Control Panel Version: 3.2.1 [Compiled: May 7th 2013]
17:20:59 [main] You are not running with administrator rights! This will work for
17:20:59 [main] most application stuff but whenever you do something with services
17:20:59 [main] there will be a security dialogue or things will break! So think
17:20:59 [main] about running this application with administrator rights!
17:20:59 [main] XAMPP Installation Directory: "c:\xampp\
17:20:59 [main] Checking for prerequisites
17:20:59 [main] All prerequisites found
17:20:59 [main] Initializing Modules
17:20:59 [main] The FileZilla module is disabled
17:20:59 [main] Starting Check-Timer
17:20:59 [main] Control Panel Ready
17:23:45 [Apache] Attempting to start Apache app...
17:23:46 [Apache] Status change detected: running
17:23:49 [mysql] Attempting to start MySQL app...
17:23:51 [mysql] Status change detected: running

Figura 42: Lanzados los servicios de Apache y MySQL

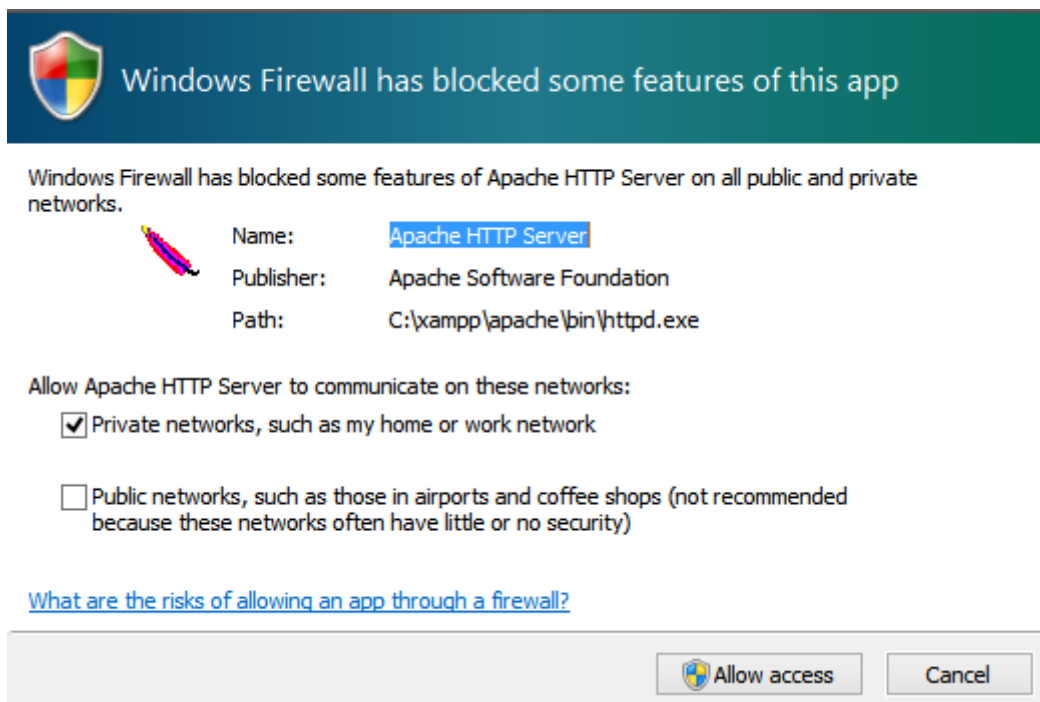


Figura 43: Detección de Firewall de Windows sobre el ejecutable httpd.exe de XAMPP

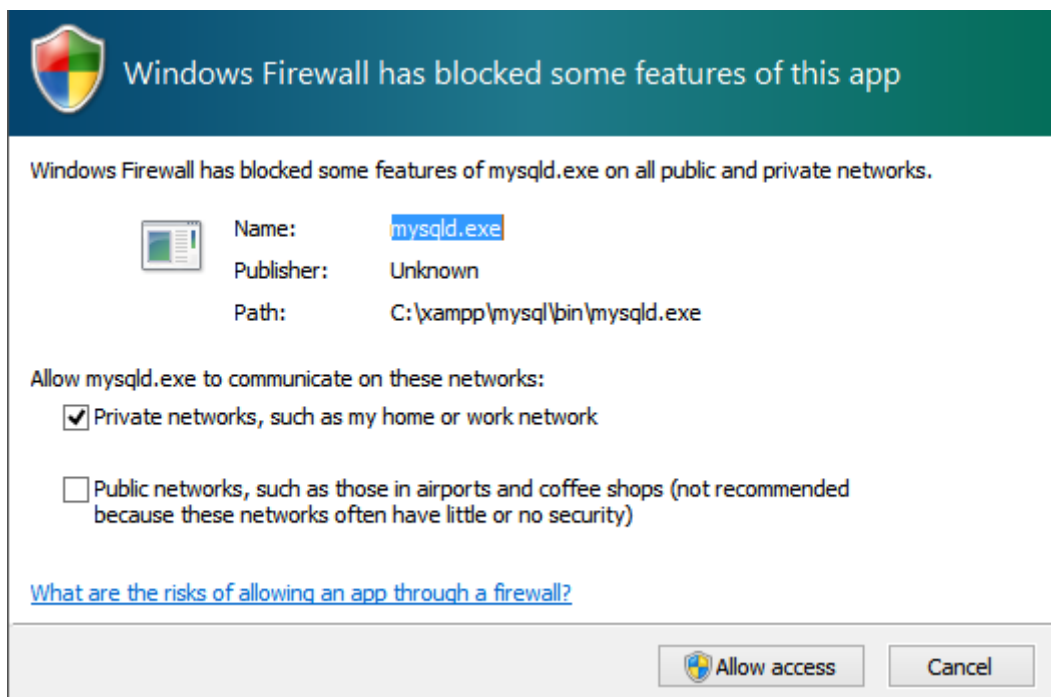


Figura 44: Detección de Firewall de Windows sobre el ejecutable mysqld.exe de XAMPP

- Abra en un navegador la página web <http://www.simaca.com> y seleccione inglés como idioma para XAMPP.

6. Seleccione la opción del menú de la barra izquierda “Security” y se abrirá una nueva página con el contenido mostrado en la [Figura 45](#), donde todos los campos no están protegidos. Para ello, se ha de ir al enlace que aparece debajo de la tabla de la [Figura 45](#) (<http://localhost/security/xamppsecurity.php>).

XAMPP SECURITY	
<i>(Requests allowed from localhost only)</i>	
This page gives you a quick overview about the security status of your XAMPP installation. (Please continue reading after the table.)	
Subject	Status
These XAMPP pages are accessible by network for everyone Every XAMPP demo page you are right now looking at is accessible for everyone over network. Everyone who knows your IP address can see these pages.	UNSECURE
The MySQL admin user root has NO password Every local user on Windows box can access your MySQL database with administrator rights. You should set a password.	UNSECURE
PhpMyAdmin is free accessible by network PhpMyAdmin is accessible by network without password. The configuration 'httpd' or 'cookie' in the "config.inc.php" can help.	UNSECURE
[XAMPP USB LITE]: Make XAMPP more safety then use this => http://localhost/security/xamppsecurity.php	

Figura 45: Contenido de la página de seguridad de XAMPP

7. En la página de configuración (ver [Figura 46](#)) de contraseñas para los servicios, deberá introducir una contraseña que sea segura.
Aviso: Recordar la contraseña de MySQL, para los accesos posteriores a la base de datos desde PHP. Una vez configurada la contraseña, debe mostrarse el mensaje como el que aparece en la [Figura 47](#). Opcional: La decisión de establecer protección del directorio XAMPP es totalmente opcional, no afecta en absoluto el funcionamiento del sistema a instalar.

Security console MySQL & XAMPP directory protection

MYSQL SECTION: "ROOT" PASSWORD

MySQL SuperUser: **root**

New password:

Repeat the new password:

PhpMyAdmin authentication: *http* ☐ *cookie* ☒

---- Security risk! ----
 Safe plain password in text file? ☐
 (File: C:\xampp\security\security\mysqlrootpasswd.txt)

XAMPP DIRECTORY PROTECTION (.htaccess)

User:

Password:

---- Security risk! ----
 Safe plain password in text file? ☐
 (File: C:\xampp\security\security\xamppdirpasswd.txt)

Figura 46: Página de configuración de contraseñas de XAMPP

Security console MySQL & XAMPP directory protection

MYSQL SECTION: "ROOT" PASSWORD

The root password was successfully changed. Please restart MYSQL for loading these changes!

MySQL SuperUser: **root**

New password:

Repeat the new password:

PhpMyAdmin authentication: *http* ☒ *cookie* ☐

Figura 47: Contraseña de MySQL configurada

8. Se pueden retirar todos los ficheros que están presentes en el directorio por defecto del contenido de la página web: “C:\xampp\htdocs” y poner en su lugar todo los ficheros del proyecto. Excepción: Los ficheros “ServidorAvisos.bat” y “ServidorWebSocketQ.bat” deben ser colocados en el directorio “C:\xampp”.
9. Se deben actualizar los datos de acceso a la base de datos en el fichero localizado en “C:\xampp\htdocs\funciones\baseDeDatos.php”

Seguridad

La instalación del parche de seguridad sobre OpenSSL se realiza siguiendo los siguientes pasos:

1. Descargar y descomprimir el documento ZIP.
2. Copiar todos ficheros incluidos en el interior de la carpeta del “xampp-opensslfix-win32” y sobrescribirlos a los que ya estaban en el directorio donde se ha instalado el servidor XAMPP: “C:\xampp\apache\bin”.

La configuración de uso obligatorio del protocolo SSL para todo el dominio del sistema, requiere la realización de los siguientes pasos (tomando como referencia el tutorial localizado en <http://robsnotebook.com/xampp-ssl-encrypt-passwords>):

1. Para poder seguir el tutorial mencionado anteriormente, se ha de modificar el fichero localizado en: “C:\xampp\apache\makecert.bat” con un editor de texto y sustituir el contenido de la línea 2 mostrado en la [Figura 48](#), por el que aparece en la [Figura 49](#), es decir, sustituir “./bin/openssl.cnf” por “./conf/openssl.cnf”.

```

1  @echo off
2  set OPENSSL_CONF=./bin/openssl.cnf
3
4  if not exist .\conf\ssl.crt mkdir .\conf\ssl.crt
5  if not exist .\conf\ssl.key mkdir .\conf\ssl.key
6
7  bin\openssl req -new -out server.csr
8  bin\openssl rsa -in privkey.pem -out server.key
9  bin\openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 365
10
11 set OPENSSL_CONF=
12 del .rnd
13 del privkey.pem
14 del server.csr
15
16 move /y server.crt .\conf\ssl.crt
17 move /y server.key .\conf\ssl.key
18
19 echo.
20 echo -----
21 echo Das Zertifikat wurde erstellt.
22 echo The certificate was provided.
23 echo.
24 pause
25

```

Figura 48: Documento original de makecert.bat

```

1  @echo off
2  set OPENSSL_CONF=./conf/openssl.cnf
3
4  if not exist .\conf\ssl.crt mkdir .\conf\ssl.crt
5  if not exist .\conf\ssl.key mkdir .\conf\ssl.key
6
7  bin\openssl req -new -out server.csr
8  bin\openssl rsa -in privkey.pem -out server.key
9  bin\openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 365
10
11 set OPENSSL_CONF=
12 del .rnd
13 del privkey.pem
14 del server.csr
15
16 move /y server.crt .\conf\ssl.crt
17 move /y server.key .\conf\ssl.key
18
19 echo.
20 echo ----
21 echo Das Zertifikat wurde erstellt.
22 echo The certificate was provided.
23 echo.
24 pause

```

Figura 49: Contenido modificado del fichero makecert.bat

2. Una vez hecho el paso 1, puede continuar con el tutorial mencionado previamente.
3. Editar el documento “httpd.conf” desde el panel de control de XAMPP, con los servicios parados. Se ha de actualizar el fichero “httpd.conf” añadiendo las siguientes líneas de código al final del fichero:

```

RewriteEngine On
RewriteCond %{HTTPS} off
RewriteRule
https://%{HTTP_HOST}%{REQUEST_URI}

```

Base de datos

La instalación de la base de datos del sistema operativo requiere la realización de los siguientes pasos:

1. Se debe editar el fichero localizado en “C:\xampp\htdocs\bdd\bdd.V.0.2.sql”, modificando el contenido de las operaciones de inserción de una cuenta por defecto que corresponde al administrador del sistema, en las líneas 190-191 y 196-197.
2. La obtención de la contraseña y el nombre de la foto del administrador, se consigue accediendo a la página “<https://localhost/bdd/cifrarDatosAdmin.php>”, con el servidor de XAMPP funcionando y los servicios lanzados.
3. El nombre de la foto del administrador se cifra a partir del DNI del administrador, con letra incluida.

4. La contraseña del administrador se muestra abiertamente, dado que el formulario no ofrece una comprobación previa de si la contraseña es la deseada. Se asume un entorno físico y digital seguro para la introducción de los datos en este formulario.

Uso

Conexión

La conexión del sistema ha de realizarse de acuerdo a las siguientes directrices:

1. Comprobar que la batería de la aeronave tenga carga suficiente.
2. Comprobar que todos los componentes de la aeronave estén bien sujetos.
3. Comprobar que las hélices están desmontadas.
4. Comprobar que esté activado WiFi en el *router* o *switch* o repetidor.
5. Iniciar el servidor XAMPP.
6. Abrir la página web principal de SIMACA y autenticarse.
7. Abrir el panel de control de XAMPP y pulsar sobre el botón “Shell”, para abrir una ventana de línea de comandos.
8. Se conecta la batería en la aeronave para asegurar su encendido.
9. Esperar a que la aeronave sincronice con los ESC y quede lista para iniciar su funcionamiento, lo cual será avisado con una serie de pitidos agudos y repetitivos.
10. Esperar de 3 a 5 segundos, tras finalizar los pitidos e ir a la ventana de línea de comandos, escribir “ServidorWebSocketQ.bat” en ella y pulsar “Enter”.
11. Esperar a que se inicie el servidor de *websocket* y se establezca la conexión socket con la aeronave.
12. Volver al navegador, esperar de 5 a 15 segundos, para que se inicie el módulo de la cámara en la aeronave.
13. Desde el menú principal de SIMACA, ir a “Manejar una Aeronave”.
14. Esperar a que se establezca la conexión *websocket* y recibir las imágenes del módulo de la cámara.
15. Pulsar las teclas correspondientes a la acción que se desea, directamente sobre la página cargada.

Precaución: No se debe mantener las teclas pulsadas, las pulsaciones han de ser cortas. La pulsación mantenida puede generar un bloqueo global en el servidor, localizado en la aeronave, lo que provoca el rechazo a nuevas peticiones o comandos emitidas por el usuario final.

Desconexión

La desconexión del sistema ha de seguir el siguiente protocolo definido:

1. Si se trata de una emergencia pulsar la tecla “Enter”, que enviará una señal de parada de emergencia a todos los motores, ajustando sus potencias a 0 directamente.
2. Se debe colocar la aeronave en posición de aterrizaje, que corresponde a una posición estable y tener los ejes de las alas paralelas a la superficie del suelo (suponiendo un terreno llano).
3. Si se trata de un aterrizaje normal y automático en un entorno seguro, se puede utilizar la función de aterrizaje, pulsando la tecla “1”.
Precaución: Durante la ejecución de esta función automática, ésta no podrá ser interrumpida de ninguna manera remotamente, hasta que no terminen de ejecutarse todas las órdenes fijadas para esta modalidad de aterrizaje.
4. Si se trata de realizar un aterrizaje normal y manual, se debe hacer descender la aeronave gradualmente pulsando para ello la tecla “h”, encargada de hacer disminuir la potencia de los motores con cada pulsación.
5. Una vez aterrizada la aeronave sobre la superficie del suelo, se debe procurar la parada total de todos los motores, pulsando la tecla de parada de emergencia “Enter”.
6. Desconectar la batería o apagar la aeronave
7. Cerrar la ventana de línea de comandos.
8. Parar los servicios de Apache y MySQL desde el panel de control de XAMPP, pulsando los botones “Stop” correspondientes.
9. Salir del panel de control de XAMPP pulsando el botón “Quit”.

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Thu Jun 05 16:19:09 CEST 2014
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)